SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 1–6

DISTRIBUTIONS OF THE EXTREME EIGENVALUES OF BETA–JACOBI RANDOM MATRICES*

IOANA DUMITRIU† AND PLAMEN KOEV‡

Abstract. We present explicit formulas for the distributions of the extreme eigenvalues of the β–Jacobi random matrix ensemble in terms of the hypergeometric function of a matrix argument. For $\beta = 1, 2, 4$, these formulas specialize to the well-known real, complex, and quaternion Jacobi ensembles, respectively.

Key words. random matrix, Jacobi distribution, MANOVA, multivariate beta, hypergeometric function of a matrix argument, eigenvalue

AMS subject classifications. 15A52, 60E05, 62H10, 65F15

DOI. 10.1137/050643234

1. Introduction.

Various multivariate statistical techniques such as canonical correlation analysis, multivariate analysis of variance, etc., are based on the distributions of the extreme eigenvalues of random matrices and, in particular, real and complex Jacobi random matrices [3, 5, 17].

The distribution of the extreme eigenvalues of the real Jacobi random matrices are well known (see section 2.3). The contribution of this paper is to generalize this result by deriving explicit expressions for the extreme eigenvalues of all β–Jacobi matrices in terms of the hypergeometric function of a matrix argument. The real, complex, and quaternion Jacobi random matrices are β–Jacobi distributed for $\beta = 1, 2$, and 4, respectively (see section 2 for the formal definitions of the classical Jacobi and β–Jacobi ensembles).

Other than the classical real case [17], the complex ($\beta = 2$) Jacobi matrices are of interest in wireless communication and signal processing [3, 5]. In general, the β–Jacobi ensembles are prominent in statistical physics in the study of the positions of the particles in a log-Coulomb gas at $2/\beta$ temperature, with Jacobi potentials [9].

The parameter β. Historically, the hypergeometric function of a matrix argument has been defined in terms of a parameter α [2, 10]. Elsewhere in random matrix theory, for example in statistical mechanics, the parameter $\beta = \frac{2}{\alpha}$ is prevalent (and known as the Boltzmann constant). This can sometimes be a source of confusion, thus we emphasize the fact that in this paper we are using the parameter β only.

Organization of the paper. In section 2 we define the Wishart, Jacobi, and β–Jacobi ensembles; we review their matrix models, define the hypergeometric function of a matrix argument, and survey the existing formulas for the distributions of the extreme eigenvalues of the real Jacobi ensemble. We present our main results—

*Received by the editors October 21, 2005; accepted for publication (in revised form) by D. Calvetti June 5, 2007; published electronically January 23, 2008.
http://www.siam.org/journals/simax/30-1/64323.html

†Department of Mathematics, University of Washington, Seattle, WA 98101 (dumitriu@math.washington.edu). This author was supported by the Miller Institute for Research in Basic Sciences, U.C. Berkeley, CA.

‡Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139 (plamen@math.mit.edu). This author was supported by National Science Foundation grant DMS–0608306.

formulas for the distributions of the extreme eigenvalues of the β–Jacobi ensemble—in section 3. We present numerical experiments in section 4.

2. Background. In this section we recall the definitions of the hypergeometric function of a matrix argument; we define the classical Jacobi random matrix ensembles, the β–Jacobi distributions and ensembles, and survey the classical relationships between them.

2.1. Hypergeometric function of a matrix argument. This function (defined below) is a series of Jack functions. The [illegible], $C_\kappa^{(\beta)}(X)$, defined for a partition κ and a symmetric matrix X, is a symmetric, homogeneous polynomial in the eigenvalues $x_1, x_2, \ldots, x_m$ of X. It generalizes the (normalized) Schur function, the zonal polynomial [20, Proposition 1.2], and the quaternion zonal polynomial to which it reduces for $\beta = 1, 2$, and 4, respectively [15].

There are several normalizations of the Jack function; in this paper we use the "C" normalization, i.e., the one for which $\sum_{\kappa \vdash k} C_\kappa^{(\beta)}(X) = (\operatorname{tr} X)^k$. The explicit definition and properties of the Jack function are available from the classical paper by Stanley [20]; we do not need them here.

DEFINITION 2.1 (hypergeometric function of a matrix argument). [illegible] $p \geq 0$ [illegible] $q \geq 0$ [illegible] X [illegible] $m \times m$ [illegible] hypergeometric function of a matrix argument X [illegible] $\beta > 0$ [illegible]

$$ {}_pF_q^{(\beta)}(a_1, \ldots, a_p; b_1, \ldots, b_q; X) \equiv \sum_{k=0}^{\infty} \sum_{\kappa \vdash k} \frac{(a_1)_\kappa^{(\beta)} \cdots (a_p)_\kappa^{(\beta)}}{k!(b_1)_\kappa^{(\beta)} \cdots (b_q)_\kappa^{(\beta)}} \cdot C_\kappa^{(\beta)}(X), $$

[illegible] $\kappa \vdash k$, [illegible] $\kappa = (\kappa_1, \kappa_2, \ldots, \kappa_m)$, $\kappa_1 \geq \kappa_2 \geq \cdots \geq \kappa_m \geq 0$, [illegible] k [illegible]

$$ (a)_\kappa^{(\beta)} \equiv \prod_{i=1}^{m} \prod_{j=1}^{\kappa_i} \left(a - \frac{\beta}{2}(i-1) + j - 1 \right) $$

[illegible] generalized Pochhammer symbol

2.2. The classical Jacobi ensembles and the β–Jacobi ensemble. The classical real, complex, and quaternion Jacobi ensembles are defined as "ratios" of real, complex, and quaternion Wishart matrices, respectively. The β–Jacobi ensembles generalize the eigenvalue distributions of the classical Jacobi ensembles, and are defined for any $\beta > 0$.

DEFINITION 2.2 (real, complex, and quaternion Wishart ensembles). [illegible] Z [illegible] $m \times n$ [illegible] $N(0, I_n \otimes \Sigma)$, $\mathcal{C}N(0, I_n \otimes \Sigma)$, $\mathcal{H}N(0, I_n \otimes \Sigma)$ [illegible] [1] [illegible] $A = Z^D Z$ [illegible] $m \times m$ [illegible] n [illegible] Σ [2] [illegible] $W_m^{(\beta)}(n, \Sigma)$, $\beta = 1, 2, 4$ [illegible]

DEFINITION 2.3 (real, complex, and quaternion Jacobi ensembles). [illegible] $A \sim W_m^{(\beta)}(n_1, \Sigma)$ [illegible] $B \sim W_m^{(\beta)}(n_2, \Sigma)$ [illegible] ($\beta = 1, 2, 4$ [illegible]) [illegible] $C = A(A+B)^{-1}$ [illegible] Jacobi [illegible]

[1] The notation $\mathcal{H}$ stands for William Hamilton, who introduced the quaternions in the 1850s.

[2] The notation Z^D stands for the quaternion conjugate transpose of the matrix Z and reduces to the Hermitian transpose Z^H when Z is complex and the transpose Z^T when Z is real.

The real Jacobi distribution is sometimes called "multivariate Beta distribution" [4, 17] and is closely related to the "MANOVA" (Multivariate ANalysis Of VAriance) distribution [17], which, rather than consider the matrix $A(A+B)^{-1}$, examines the matrix AB^{-1}.

REMARK 2.4.

PROPOSITION 2.5. λ C $\lambda \in [0,1]$ If λ is an eigenvalue of C, then there exists an eigenvalue $\tilde{\lambda}$ of AB^{-1} such that $\lambda = \tilde{\lambda}/(1+\tilde{\lambda})$. Since A and B are positive semidefinite, $\tilde{\lambda}$ is positive as a generalized eigenvalue of the matrix pair (A, B), and the desired conclusion follows. □

The following definition is central to this paper.

DEFINITION 2.6 (β–Jacobi ensembles). $\beta > 0$ $a_1, a_2 > \frac{\beta}{2}(m-1)$ β

$$\frac{1}{\mathcal{I}(m,\beta,a_1,a_2)} \prod_{i=1}^{m} \lambda_i^{a_1 - \frac{\beta}{2}(m-1)-1} (1-\lambda_i)^{a_2 - \frac{\beta}{2}(m-1)-1} \prod_{i<j} |\lambda_i - \lambda_j|^{\beta}, \tag{2.1}$$

$$\mathcal{I}(m,\beta,a_1,a_2) = \frac{\Gamma_m^{(\beta)}\big(1+\frac{\beta}{2}m\big)}{\pi^{\frac{m(m-1)\beta}{2}} \Big(\Gamma\big(1+\frac{\beta}{2}\big)\Big)^m} \cdot \frac{\Gamma_m^{(\beta)}(a_1)\,\Gamma_m^{(\beta)}(a_2)}{\Gamma_m^{(\beta)}(a_1+a_2)}$$

[19]

$$\Gamma_m^{(\beta)}(c) \equiv \pi^{\frac{m(m-1)\beta}{4}} \prod_{i=1}^{m} \Gamma\left(c - \frac{\beta}{2}(i-1)\right) \qquad \Re(c) > \frac{\beta}{2}(m-1)$$

multivariate gamma function $\beta > 0$

From Theorem 3.3.4 in Muirhead [17], the real Jacobi matrices are β–Jacobi distributed for $\beta = 1$. By repeating the same argument for $\beta = 2$ and 4 we obtain the following theorem.

THEOREM 2.7. 2.3 C β $a_1 = \frac{\beta}{2}n_1$ $a_2 = \frac{\beta}{2}n_2$ $\beta = 1, 2$ 4

Following the methods of Dumitriu and Edelman [7], several matrix models were proposed that have β–Jacobi distributions [11, 14, 21]; here we present the one from Sutton [21].

THEOREM 2.8. $J \equiv Z^T Z$ β

$$Z \equiv \begin{bmatrix} c_m & -s_m c'_{m-1} & & \\ & c_{m-1} s'_{m-1} & & \\ & & & -s_2 c'_1 \\ & & & c_1 s'_1 \end{bmatrix},$$

$$c_k \sim \sqrt{\mathrm{Beta}\big(a_1 - \tfrac{\beta}{2}(m-k), a_2 - \tfrac{\beta}{2}(m-k)\big)}, \quad s_k = \sqrt{1-c_k^2},$$
$$c'_k \sim \sqrt{\mathrm{Beta}\big(\tfrac{\beta}{2}k, a_1 + a_2 - \tfrac{\beta}{2}(2m-k-1)\big)}, \quad s'_k = \sqrt{1-c_k'^2},$$

Beta

2.3. Survey of the real case ($\beta = 1$). We now survey the well-known distribution of the largest eigenvalue of the real Jacobi matrix (which is β–Jacobi distributed for $\beta = 1$). We generalize this result to any β in section 3.

Let $A \sim W_m^{(1)}(n_1, \Sigma)$, $B \sim W_m^{(1)}(n_2, \Sigma)$, where $n_1 \geq m$ and $n_2 \geq m$, be independent real Wishart matrices. For the distribution of the largest eigenvalue $\lambda_{\max}$ of the real Jacobi matrix $A(A+B)^{-1}$ we have [4, equation (61)]:

$$P(\lambda_{\max} < x) = \frac{\Gamma_m^{(1)}(\frac{n_1+n_2}{2})\Gamma_m^{(1)}(\frac{m+1}{2})}{\Gamma_m^{(1)}(\frac{n_1+m+1}{2})\Gamma_m^{(1)}(\frac{n_2}{2})} \cdot x^{\frac{mn_1}{2}} \cdot {}_2F_1^{(1)}(\tfrac{n_1}{2}, \tfrac{-n_2+m+1}{2}; \tfrac{n_1+m+1}{2}; xI).$$

Using [1, equation (9)] we obtain an explicit expression for the of $\lambda_{\max}$:

$$\begin{aligned}\operatorname{dens}(\lambda_{\max}) = \frac{mn_1}{2} \cdot \frac{\Gamma_m^{(1)}(\frac{n_1+n_2}{2})\Gamma_m^{(1)}(\frac{m+1}{2})}{\Gamma_m^{(1)}(\frac{n_1+m+1}{2})\Gamma_m^{(1)}(\frac{n_2}{2})} \cdot (1-x)^{\frac{n_2-m-1}{2}} x^{\frac{mn_1}{2}-1} \\ \times {}_2F_1^{(1)}\big(\tfrac{n_1-1}{2}, \tfrac{m-n_2+1}{2}; \tfrac{n_1+m+1}{2}; xI_{m-1}\big).\end{aligned}$$

3. The extreme eigenvalues of the β–Jacobi ensembles. We obtain our main result in this paper by integrating the joint eigenvalue density of a β–Jacobi matrix and expressing the resulting integral in terms of the hypergeometric function of a matrix argument.

The connection with ${}_2F_1^{(\beta)}$ comes from Kaneko [10, Theorem 5]:

$$\begin{aligned}(3.1)\quad & {}_2F_1^{(\beta)}(r, a, a+b, tI_m) \\ & = \frac{1}{\mathcal{I}(m, \beta, a, b)} \int_{[0,1]^m} \prod_{i=1}^m x_i^{\lambda_1}(1-x_i)^{\lambda_2}(1-tx_i)^{-r} \prod_{i<j} |x_j - x_i|^\beta dx_1 \cdots dx_m,\end{aligned}$$

where $a = \lambda_1 + \frac{\beta}{2}(m-1) + 1$ and $b = \lambda_2 + \frac{\beta}{2}(m-1) + 1$.

Now let the $m \times m$ matrix J_β be β–Jacobi distributed with parameters a_1, a_2, and let $\lambda_{\max}$ and $\lambda_{\min}$ be its largest and smallest eigenvalues, respectively.

THEOREM 3.1. $\lambda_{\max}$ $\lambda_{\min}$

$$\begin{aligned}P(\lambda_{\max} < x) &= \frac{\Gamma_m^{(\beta)}(a_1+a_2) \cdot \Gamma_m^{(\beta)}\big(\frac{\beta}{2}(m-1)+1\big)}{\Gamma_m^{(\beta)}\big(a_1 + \frac{\beta}{2}(m-1)+1\big) \cdot \Gamma_m^{(\beta)}(a_2)} \cdot x^{ma_1} \\ &\quad \times {}_2F_1^{(\beta)}\left(a_1, \frac{\beta}{2}(m-1)+1-a_2; a_1 + \frac{\beta}{2}(m-1)+1; xI_m\right), \qquad (3.2) \\ P(\lambda_{\min} < x) &= 1 - \frac{\Gamma_m^{(\beta)}(a_1+a_2) \cdot \Gamma_m^{(\beta)}\big(\frac{\beta}{2}(m-1)+1\big)}{\Gamma_m^{(\beta)}\big(a_2 + \frac{\beta}{2}(m-1)+1\big) \cdot \Gamma_m^{(\beta)}(a_1)} \cdot (1-x)^{ma_2} \\ &\quad \times {}_2F_1^{(\beta)}\left(a_2, \frac{\beta}{2}(m-1)+1-a_1; a_2 + \frac{\beta}{2}(m-1)+1; (1-x)I_m\right).\end{aligned}$$

. We start with the joint eigenvalue density of J_β, (2.1), and note that to compute the distribution of the largest eigenvalue of J_β we need to integrate this density from 0 to xI_m. For $b_i \equiv a_i - \frac{\beta}{2}(m-1) - 1$, $i = 1, 2$, we have

$$P(J_\beta < xI_m) = \frac{1}{\mathcal{I}(m, \beta, a_1, a_2)} \int_{[0,x]^m} \prod_{i=1}^m \lambda_i^{b_1}(1-\lambda_i)^{b_2} \prod_{i<j} |\lambda_i - \lambda_j|^\beta d\lambda_1 \cdots d\lambda_m.$$

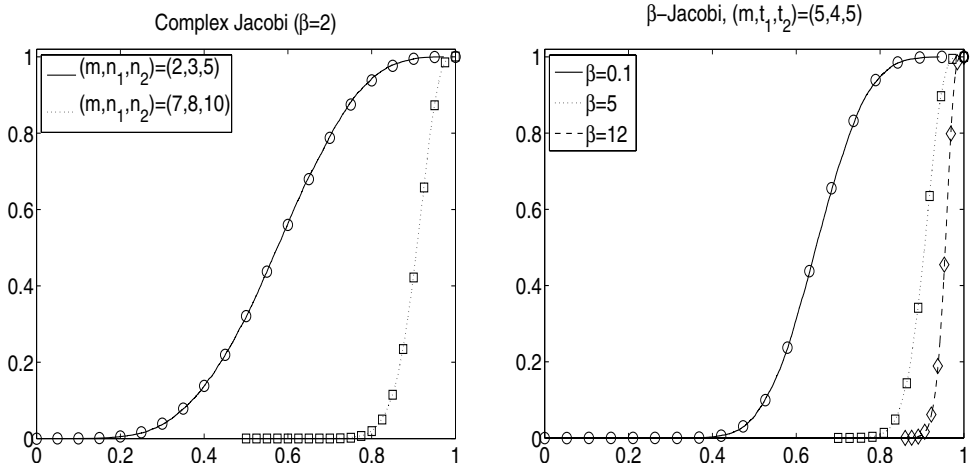


FIG. 1. *Left: Distributions of* $\lambda_{\max}$ *of a complex Jacobi matrix for two sets of parameters* (m, n_1, n_2). *The solid and dotted lines are the empirical distributions with* 10,000 *replications; "*$\circ$*" and "*$\square$*" are the corresponding analytical predictions* (3.2). *Right: Distributions of* $\lambda_{\max}$ *of a* 5×5 β*–Jacobi matrix with* $a_i = t_i + \frac{\beta}{2}(m-1) + 1, i = 1, 2$ *for different values of* β. *The solid, dotted, and dashed lines are the empirical distributions with* 10,000 *replications; "*$\circ$*," "*$\square$*," and "*$\diamond$*" are the corresponding analytical predictions* (3.3).

We make an m-dimensional change of variables $x\tilde{\lambda}_i = \lambda_i$ to obtain

$$P(J_\beta < xI_m) = \frac{x^{ma_1}}{\mathcal{I}(m, \beta, a_1, a_2)} \int_{[0,1]^m} \prod_{i=1}^{m} \tilde{\lambda}_i^{b_1} (1 - x\tilde{\lambda}_i)^{b_2} \prod_{i<j} |\tilde{\lambda}_i - \tilde{\lambda}_j|^\beta d\tilde{\lambda}_1 \cdots d\tilde{\lambda}_m.$$

We evaluate the last integral using (3.1) and get (3.2).

The result for $\lambda_{\min}$ follows immediately by observing that $1 - \lambda_{\min}$ is the largest eigenvalue of $I - J_\beta$, which is β–Jacobi distributed with parameters a_2 and a_1. ∎

When $t \equiv a_2 - \frac{\beta}{2}(m-1) - 1$ is a nonnegative integer, the hypergeometric series in (3.2) terminates and becomes a polynomial of degree mt. Then we can use Proposition 11.47 in Forrester [9] to obtain the expression

$$P(\lambda_{\max} < x) = x^{ma_1} \sum_{k=0}^{mt} \sum_{\kappa \vdash k,\, \kappa_1 \le t} \frac{1}{k!} (a_1)_\kappa^{(\beta)} C_\kappa^{(\beta)}((1-x)I), \tag{3.3}$$

which numerically is often much more feasible than (3.2).

Analogous results extending the well-known distributions of the extreme eigenvalues of real random matrices [4, 17] were obtained for β–Laguerre ensembles in [6, section 10.2] and complex Wishart ensembles in [18].

4. Numerical experiments. We performed extensive numerical tests against Monte-Carlo experiments to confirm the correctness of Theorem 3.2. We report the results of two tests whose results were typical.

In our first experiment we tested formula (3.2) against the empirical distribution of the largest eigenvalue of a complex Jacobi matrix. The more samples we used for the empirical distribution, the better approximation it was to the analytical prediction, with 10,000 samples sufficing for a perfect visual match. We generated our sample matrices in MATLAB [16] as C=A/(A+B), where A and B were complex

Wishart matrices generated as A=Z'*Z/2, where Z=randn(n_1, m)+i*randn(n_1, m), with B generated analogously. We evaluated the analytical formula (3.2), also in MATLAB, using the algorithms for computing ${}_pF_q^{(\beta)}$ from [12, 13]. We plot the results in Figure 1, left.

In our second experiment we demonstrate the β dependence of the largest eigenvalue of a 5×5 β–Jacobi matrix with parameters $a_i = t_i + \frac{\beta}{2}(m-1)+1, i = 1, 2$, where we fixed $t_1 = 4$ and $t_2 = 5$. In Figure 1, right, we plotted the empirical distribution from 10,000 replications which, again, matched the theoretical prediction (3.3) (which we could use since $a_2 - \frac{\beta}{2}(m-1) - 1 = t_2 = 5$ was a nonnegative integer).

In line with the results of [8], this experiment supports a conjecture that as β increases (and so do a_1 and a_2), the largest eigenvalue of the β–Jacobi ensemble approaches $1 = \lim_{a,b \to -1} \lambda^{a,b}$, where $\lambda^{a,b}$ is the largest root of the mth orthogonal Jacobi polynomial $J_m^{a,b}(x)$.

Acknowledgments. We thank Alan Edelman and Iain Johnstone for many illuminating discussions during the preparation of this paper.

REFERENCES

[1] P.-A. Absil, A. Edelman, and P. Koev, *On the largest principal angle between random subspaces*, Linear Algebra Appl., 414 (2006), pp. 288–294.
[2] T. Baker and P. Forrester, *The Calogero–Sutherland model and generalized classical polynomials*, Comm. Math. Phys., 188 (1997), pp. 175–216.
[3] P. Chen, G. Genello, and M. Wicks, *Estimating the number of signals in presence of colored noise*, in Proceedings of the IEEE Radar Conference 2004, Philadelphia, PA, 2004, pp. 432–437.
[4] A. G. Constantine, *Some non-central distribution problems in multivariate analysis*, Ann. Math. Statist., 34 (1963), pp. 1270–1285.
[5] D. M. Dlugos and R. A. Scholtz, *Acquisition of spread spectrum signals by an adaptive array*, IEEE Transactions on Acoustics, Speech and Signal Processing, 37 (1989), pp. 1253–1270.
[6] I. Dumitriu, *Eigenvalue Statistics for the Beta-Ensembles*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2003.
[7] I. Dumitriu and A. Edelman, *Matrix models for beta-ensembles*, J. Math. Phys., 43 (2002), pp. 5830–5847.
[8] I. Dumitriu and A. Edelman, *Eigenvalues of Hermite and Laguerre ensembles: Large beta asymptotics*, Ann. Inst. H. Poincaré Probab. Statist., 41 (2005), pp. 1083–1099.
[9] P. Forrester, *Log-gasses and Random matrices*, available online at http://www.ms.unimelb.edu.au/~matpjf/matpjf.html.
[10] J. Kaneko, *Selberg integrals and hypergeometric functions associated with Jack polynomials*, SIAM J. Math. Anal., 24 (1993), pp. 1086–1110.
[11] R. Killip and I. Nenciu, *Matrix models for circular ensembles*, Int. Math. Res. Not., 50 (2004), pp. 2665–2701.
[12] P. Koev, *Software for Computing the Hypergeometric Function of a Matrix Argument*, available online at http://www-math.mit.edu/~plamen.
[13] P. Koev and A. Edelman, *The efficient evaluation of the hypergeometric function of a matrix argument*, Math. Comp., 75 (2006), pp. 833–846.
[14] R. A. Lippert, *A matrix model for the β-Jacobi ensemble*, J. Math. Phys., 44 (2003), pp. 4807–4816.
[15] I. G. Macdonald, *Symmetric Functions and Hall Polynomials*, 2nd ed., Oxford University Press, New York, 1995.
[16] The MathWorks, Inc., *MATLAB Reference Guide*, Natick, MA, 1992.
[17] R. J. Muirhead, *Aspects of Multivariate Statistical Theory*, John Wiley & Sons Inc., New York, 1982.
[18] T. Ratnarajah, R. Vaillancourt, and M. Alvo, *Eigenvalues and condition numbers of complex random matrices*, SIAM J. Matrix Anal. Appl., 26 (2004/05), pp. 441–456.
[19] A. Selberg, *Remarks on a multiple integral*, Norsk Mat. Tidsskr., 26 (1944), pp. 71–78.
[20] R. P. Stanley, *Some combinatorial properties of Jack symmetric functions*, Adv. Math., 77 (1989), pp. 76–115.
[21] B. Sutton, *The Stochastic Operator Approach to Random Matrix Theory*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2003.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 7–40

RIGIDITY IN FINITE-ELEMENT MATRICES: SUFFICIENT CONDITIONS FOR THE RIGIDITY OF STRUCTURES AND SUBSTRUCTURES*

GIL SHKLARSKI† AND SIVAN TOLEDO†

Abstract. We present an algebraic theory of rigidity for finite-element matrices. The theory provides a formal algebraic definition of finite-element matrices; notions of rigidity of finite-element matrices and of mutual rigidity between two such matrices; and sufficient conditions for rigidity and mutual rigidity. We also present a novel sparsification technique, called *fretsaw extension*, for finite-element matrices. We show that this sparsification technique generates matrices that are mutually rigid with the original matrix. We also show that one particular construction algorithm for fretsaw extensions generates matrices that can be factored with essentially no fill. This algorithm can be used to construct preconditioners for finite-element matrices. Both our theory and our algorithms are applicable to a wide range of finite-element matrices, including matrices arising from finite-element discretizations of both scalar and vector partial differential equations (e.g., electrostatics and linear elasticity). Both the theory and the algorithms are purely algebraic-combinatorial. They manipulate only the element matrices and are oblivious to the geometry, the material properties, and the discretization details of the underlying continuous problem.

Key words. rigidity, finite elements, element matrices, null spaces, combinatorial preconditioners, support preconditioners

AMS subject classifications. 65F10, 65F05, 65F50, 65Y20, 05C50, 05C85, 52C25

DOI. 10.1137/060650295

1. Introduction.

This paper presents an algebraic-combinatorial theory of rigidity for finite-element matrices and applies this theory to two important problems: determining whether a finite-element matrix represents a rigid structure, and determining whether a matrix representing a structure and a matrix representing a substructure have the same range and null space. The paper addresses these problems by providing simple sufficient conditions for rigidity and null-space equality, and by providing linear-time algorithms (assuming bounded element degrees) to test these conditions.

Our results employ three new technical tools—one combinatorial and two algebraic. One algebraic tool is a purely algebraic definition of the rigidity relationships between two rank-deficient matrices.[1] The other algebraic tool is a definition of a finite-element matrix A as a sum of symmetric semidefinite matrices $\{A_e\}_{e=1}^{k}$ that all satisfy a certain condition. The combinatorial tool is a graph, called the [illegible], that represents the rigidity relationships between the terms A_e of a finite-element matrix $A = \sum_e A_e$. These tools may be applicable to the solution of other problems involving finite-element matrices.

*Received by the editors January 18, 2006; accepted for publication (in revised form) by E. Ng June 18, 2007; published electronically January 23, 2008. This research was supported by an IBM Faculty Partnership Award, by grant 848/04 from the Israel Science Foundation (founded by the Israel Academy of Sciences and Humanities), and by grant 2002261 from the United States–Israel Binational Science Foundation.

http://www.siam.org/journals/simax/30-1/65029.html

†School of Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel (shagil@tau.ac.il, stoledo@tau.ac.il).

[1]The literature contains another definition of matrix rigidity, originally defined by Valiant [20]. Our definition is completely different and unrelated.

The concept of rigidity is usually associated with elastic structures and with finite-element models of such structures. An elastic structure is rigid if any deformation of it that is not a translation and/or rotation requires energy. A coin is rigid; a door hinge is not. Our theory of rigidity is consistent with the traditional concept of rigidity, but it is purely algebraic and more general. By purely algebraic, we mean that our theory uses only the element matrices A_e and a basis for the rigid body motions (e.g., translations and rotations) of the structure. Our theory and algorithms do not depend on the geometry of the structure or on the details of the finite-element discretization. Our theory generalizes the concept of rigidity in a natural way from finite-element models of elastic structures to models of other physical systems, such as electrostatics.

On the other hand, our theory provides only sufficient conditions for rigidity. Characterizing rigidity exactly is difficult, even if we limit our attention to specific families of elastic structures. Consider, for example, a structure consisting of struts (elastic bars) connected at their endpoints by pin joints. The struts can only elongate or compress, and the struts connected to a pin are free to rotate around the pin. The rigidity of such structures in two dimensions has been extensively studied and is now well understood. However, the conditions that characterize the rigidity of a two-dimensional structure are expensive to check [14], and they do not generalize easily to three-dimensional trusses and to other structures. Our theory of rigidity avoids these difficulties by focusing on characterizations that are simple and general but only sufficient. In fact, structures consisting of struts always fail our sufficient conditions.

Our new theory is essentially an algebraic-combinatorial characterization of finite-element models of structures that are, informally speaking, "evidently rigid." Models of structures that are rigid due to complex nonlocal interactions between parts of the structure will usually fail our conditions. The main contributions of this paper are formal and easily computed characterizations of "evidently rigid" structures. We, therefore, call structures that pass our test [illegible]. We apply these characterizations to the construction of algorithms that find certain minimally rigid substructures of a rigid structure.

The results in this paper are a step toward the generalization of results in spectral graph theory from Laplacians to finite-element matrices. We are particularly interested in an area of spectral graph theory called [illegible] or [illegible]. This area is mostly concerned with constructing an approximation B to a matrix A in three steps: (1) building a graph G_A that represents A, (2) approximating G_A by a simpler graph G_B, and (3) building the matrix B that corresponds to G_B. The graph G_B should be simpler in some way than G_A (e.g., smaller balanced vertex separators), and the generalized eigenvalues λ of $Ax = \lambda Bx$ should not be very large or very small. Much progress has been made in this area, but only when A is a Laplacian [2, 11, 18, 19, 5], is a diagonally dominant symmetric matrix (i.e., G_A is a signed graph) [4, 11], or can be well approximated by a Laplacian [6].

This paper makes three contributions to support preconditioning of finite-element matrices. First, the paper provides a reasonable definition of what a finite-element matrix is: a sum of element matrices whose null spaces are derived from a single global null space. Second, the paper provides a graph model of finite-element matrices and proposes graph algorithms for sparsifying the coefficient matrix A. Third, the paper provides simple combinatorial conditions that allow us to show that the range and null space of the sparsified matrix (the preconditioner) B are the same as those of A. The qualitative range and null-space equalities are weaker statements than quan-

titative bounds on the generalized eigenvalues, but they are a step toward eigenvalue bounds. A weighted rigidity graph may allow us to bound eigenvalues and generalized eigenvalues. The same technical tools may also be applicable to the generalization of other results in spectral graph theory, such as Cheeger-type bounds [7, 9, 1].

We use these three contributions to define algebraic methods to sparsify finite-elements matrices. One method drops elements from the structure. The other method, called fretsaw preconditioning, cuts some of the connections between elements. The fretsaw preconditioning algorithm that we present in the paper constructs, using an almost linear amount of work, a preconditioner that can be applied in linear time in every iteration and which has the same null space as the original matrix (the linear cost is with respect to the number of unknowns assuming a fixed element degree). The construction is purely algebraic—the same algorithm, data structure, and code work on elasticity and electrostatics in two and three dimensions.

The paper is quite technical and fairly complex. It may seem strange that all of this complexity is needed to prove results that are physically intuitive. If a structure is evidently rigid, why is all the algebraic and notational complexity needed? The answer appears to be that the complexity is a result of our insistence on a purely algebraic and combinatorial analysis. We do not rely directly on any physical or continuous properties of the structures that we analyze. Our analysis reaches physically intuitive conclusions, but the algebraic path toward these conclusions is complex. We believe that the generality and software-engineering advantages of a purely algebraic approach are worth the complexity of the paper. Furthermore, the analysis is complex, but the algorithms that we propose are both general and simple.

The paper is organized as follows. Finite-element matrices are sums of very sparse terms called element matrices. Most of the rows and columns in each element matrix contain only zeros. Such matrices have a trivial null space that the zero columns generate and sometimes another null subspace that is more interesting. Our study of rigidity is essentially a study of these nontrivial subspaces. Section 2 defines these subspaces and analyzes their properties. The combinatorial structure that we use, the rigidity graph, is defined by rigidity relationships between pairs of element matrices. These relationships are defined and explored in sections 3 and 4. One of our ultimate goals in this paper is to show that a connected rigidity graph implies that the underlying structure is rigid. Unfortunately, this is not true for collections of arbitrary element matrices; they must have something in common for their rigidity graph to be useful. This common property is called null-space compatibility. Its definition and significance are explained in section 5. The rigidity graph itself is defined in section 6, along with a proof that a connected rigidity graph implies the rigidity of the structure. Section 7 studies three families of finite-element matrices and their rigidity graphs to further illustrate the concepts presented earlier. In section 8 we present two methods for sparsifying a finite-element matrix while preserving its null space. The more sophisticated method, called spanning-tree fretsaw extension, always leads to simplified finite-element matrices that can be factored with essentially no fill. We present four numerical examples of the use of spanning-tree fretsaw extension as preconditioners in section 9. We conclude the paper with a few open problems in section 10.

2. The essential null space of a matrix. Rigidity is closely related to relationships between null spaces. We therefore start our analysis with definitions and lemmas concerning the null space of matrices with zero columns.

DEFINITION 2.1. *Let A be an $m \times n$ matrix, and let $\mathcal{Z}_A \subseteq \{1,\ldots,n\}$ be the largest set of column indices such that there exists a vector x with the following properties. The set $\mathcal{N}_A = \{1,\ldots,n\} \setminus \mathcal{Z}_A$ is called the essential null space of A, and x is called a*

- $Ax = 0$,
- $x_i = 0$, $i \in \mathcal{Z}_A$.

The trivial null space of A is the set $\mathcal{Z}_A$, and a vector x that satisfies $x_i = 0$, $i \in \mathcal{N}_A$ is called an essential null vector, or enull(A), and a trivial null vector, or tnull(A).

Clearly, the essential and trivial null spaces of a matrix are orthogonal, and their span (union) is simply the null space of the matrix.

DEFINITION 2.2. *The restriction of a vector y to the set $\mathcal{N}_A$ is the vector*

$$x_i = \begin{cases} y_i, & i \in \mathcal{N}_A, \\ 0 & \text{otherwise.} \end{cases}$$

(the vector has the same dimension as y) is called the extension with respect to $\mathcal{N}_A$ of x if $x_i = 0$, $i \in \mathcal{Z}_A$, and y is the extension of $y_i = x_i$, $i \in \mathcal{N}_A$.

LEMMA 2.3. *Let y be the extension with respect to $\mathcal{N}_A$ of $x \in \text{enull}(A)$. Then $y \in \text{null}(A)$.*

Proof. Let $z = y - x$, so $y = x + z$. Since $y_i = x_i$ for $i \in \mathcal{N}_e$, we have $z_i = 0$ for $i \in \mathcal{N}_A$. Therefore, $Az = 0$, so $Ay = Ax + Az = 0 + 0 = 0$. ∎

LEMMA 2.4. *Let $y \in \text{null}(A)$. Then $\mathcal{N}_A$ is the union of the supports of two vectors in enull(A).*

Proof. Let $y = y_N + y_Z$ be a splitting of y into a vector y_N with nonzeros only in $\mathcal{N}_A$ and y_Z with nonzeros only in $\mathcal{Z}_A$. Clearly, $Ay_N = Ay_Z = 0$. The result follows from the fact that y_N is the restriction of y to $\mathcal{N}_A$. ∎

LEMMA 2.5. *Let A and B be $n \times n$ matrices such that A is SPSD and $A + B$ is SPSD. Then $\text{null}(A + B) = \text{null}(A) \cap \text{null}(B)$.*

Proof. Let $x \in \text{null}(A + B)$ and suppose for contradiction that $Ax \neq 0$. A has a decomposition $A = LL^T$. Since $Ax \neq 0$, we also have $L^T x \neq 0$, so $x^T LL^T x = x^T Ax > 0$. Therefore, $x^T Bx = x^T (A + B)x - x^T Ax = 0 - x^T Ax < 0$, which is a contradiction. Therefore, $x \in \text{null}(A)$ and, similarly, $x \in \text{null}(B)$. This shows that $\text{null}(A + B) \subseteq \text{null}(A) \cap \text{null}(B)$. The other direction is trivial. ∎

A column that is nonzero in both A and B can be a zero in $A + B$ due to cancellation. The next lemma shows that this cannot happen when the terms are SPSD matrices.

LEMMA 2.6. *Let A and B be $n \times n$ SPSD matrices. Then $\mathcal{N}_{A+B} = \mathcal{N}_A \cup \mathcal{N}_B$.*

Proof. Clearly $\mathcal{N}_{A+B} \subseteq \mathcal{N}_A \cup \mathcal{N}_B$. Suppose for contradiction that the lemma does not hold. Then there is a column index j in $\mathcal{N}_A$ or in $\mathcal{N}_B$ that is not in $\mathcal{N}_{A+B}$. Without loss of generality assume that $j \in \mathcal{N}_A$. Let x be the jth unit vector. Since $j \in \mathcal{N}_A$, Ax, which is simply the jth column of A, is nonzero. But since $j \notin \mathcal{N}_{A+B}$, we also have $(A + B)x = 0$, which is a contradiction to Lemma 2.5. ∎

The last lemma in this section shows the relationship between null-space containment and the sets $\mathcal{N}$ and $\mathcal{Z}$.

LEMMA 2.7. *Let A be an $m \times n$ matrix, and let B be an $l \times n$ matrix. If $\text{null}(B) \subseteq \text{null}(A)$ then $\mathcal{Z}_B \subseteq \mathcal{Z}_A$ and $\mathcal{N}_A \subseteq \mathcal{N}_B$.*

Proof. Let $j \in \mathcal{Z}_B$ and let e_j be the jth unit vector. By definition, $Be_j = 0$. By the assumption on the null spaces, $Ae_j = 0$. This implies that $j \in \mathcal{Z}_A$. Therefore, $\mathcal{Z}_B \subseteq \mathcal{Z}_A$, so the complements of these sets satisfy $\mathcal{N}_A \subseteq \mathcal{N}_B$. ∎

3. Rigidity relationships. This section introduces the main notion of this paper: rigidity relationships.

DEFINITION 3.1. [illegible] A [illegible] m [illegible] n [illegible] A [illegible] B [illegible] ℓ [illegible] n [illegible] $y \in \text{enull}(B)$ [illegible] mapping [illegible] $x \in \text{enull}(A)$ [illegible] $y_i = x_i$ [illegible] $i \in \mathcal{N}_A \cap \mathcal{N}_B$ [illegible] rigid [illegible] x [illegible] A [illegible] rigid with respect to [illegible] B [illegible] $x \in \text{enull}(A)$ [illegible] $y \in \text{enull}(B)$ [illegible] mutually rigid [illegible]

3.2. Mutual rigidity does not follow automatically from one-sided rigidity. Consider, for example,

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}.$$

A is rigid with respect to B, because vectors in $\text{enull}(A)$ have the form $\begin{bmatrix} 0 & \alpha & \alpha & 0 \end{bmatrix}^T$, and they have a unique extension to vectors in $\text{enull}(B)$, namely, $\begin{bmatrix} \alpha & \alpha & \alpha & \alpha \end{bmatrix}^T$. But vectors in $\text{enull}(B)$, which have the form $\begin{bmatrix} \alpha & \alpha & \beta & \beta \end{bmatrix}^T$, are not in $\text{null}(A)$ unless $\alpha = \beta$.

3.3. Let $A_1 = \begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix}$ and $A_2 = \begin{bmatrix} 0 & 1 & -1 & 0 \end{bmatrix}$. The two matrices are mutually rigid. We have

$$\text{enull}\,(A_1) = \text{span}\begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T,$$

$$\text{enull}\,(A_2) = \text{span}\begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^T.$$

Therefore, for every $x = \begin{bmatrix} \alpha & \alpha & 0 & 0 \end{bmatrix}^T \in \text{enull}(A_1)$, there is a unique $y = \begin{bmatrix} 0 & \alpha & \alpha & 0 \end{bmatrix}^T \in \text{enull}(A_1)$, and symmetrically for A_2.

Now let $A_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$ and $A_4 = \begin{bmatrix} 0 & 0 & -1 & 1 \end{bmatrix}$. A_1 is not rigid with respect to either of these two. It is not rigid with respect to A_3 because $\text{enull}(A_3) = \{0\}$, so for an $x \in \text{enull}(A_1)$ there is no rigid y in $\text{enull}(A_3)$. A_1 is not rigid with respect to A_4 because for $x = \begin{bmatrix} \alpha & \alpha & 0 & 0 \end{bmatrix}^T \in \text{enull}(A_1)$, any $y = \begin{bmatrix} 0 & 0 & \beta & \beta \end{bmatrix}^T$ is in $\text{enull}(A_4)$, so the mapping is not unique.

We now show how to test whether a matrix A is rigid with respect to another matrix B. For an m-by-n matrix A, we define Ψ_A and $\Psi_{\bar{A}}$ to be the n-by-n diagonal matrices

$$\left[\Psi_A\right]_{jj} = \begin{cases} 1, & j \in \mathcal{N}_A, \\ 0, & j \in \mathcal{Z}_A, \end{cases} \quad \text{and} \quad \left[\Psi_{\bar{A}}\right]_{jj} = \begin{cases} 0, & j \in \mathcal{N}_A, \\ 1, & j \in \mathcal{Z}_A\,. \end{cases}$$

For two matrices A and B with n columns each, we define $\Psi_{A,B}$ to be the n-by-n diagonal matrix

$$\left[\Psi_{A,B}\right]_{jj} = \begin{cases} 1, & j \in \mathcal{N}_A \cap \mathcal{N}_B, \\ 0 & \text{otherwise.} \end{cases}$$

Let x be a vector in $\text{enull}(A)$. If x has a mapping to $y \in \text{enull}(B)$, then y must

satisfy the equations

$$\Psi_{\bar{B}} y = 0,$$

$$By = 0,$$

$$\Psi_{A,B} y = \Psi_{A,B} x.$$

The first two conditions constrain y to be in enull(B) and the third condition constrains y to be a mapping of x. If this linear system is inconsistent, then x has no rigid mapping to $y \in$ enull(B), so A is not rigid with respect to B. Even if the system is consistent for all $x \in$ enull(A), A is not necessarily rigid with respect to B. If the coefficient matrix $R_{A,B} = \begin{bmatrix} \Psi_{\bar{B}}^T & B^T & \Psi_{A,B}^T \end{bmatrix}^T$ is rank deficient, the mappings are not unique.

Therefore, to test rigidity we must check that for all $x \in$ enull(A), the vector $\begin{bmatrix} 0 & 0 & x^T \Psi_{A,B}^T \end{bmatrix}^T$ is spanned by the columns of $R_{A,B}$ and that the columns of $R_{A,B}$ are linearly independent. We now derive equivalent conditions, but on a much smaller system. First, we drop rows and columns $\mathcal{Z}_B$ from the coefficient matrix and rows $\mathcal{Z}_B$ from y. These rows correspond to equations that constrain $y_i = 0$ for $i \in \mathcal{Z}_B$. Since these elements of y are not used in any of the other equations, we can drop them without making an inconsistent system into a consistent one. Also, these columns are linearly independent, and all the other columns are independent of them. Therefore, dropping these $|\mathcal{Z}_B|$ rows and columns reduces the rank of $R_{A,B}$ by exactly $|\mathcal{Z}_B|$; therefore, $R_{A,B}$ is full rank if and only if the remaining rows and columns form a full-rank matrix. Now we drop all the zero rows from the system—rows $\mathcal{N}_B$ in the $\Psi_{\bar{B}}$ block of $R_{A,B}$, the zero rows from the B block, and the zero rows from the $\Psi_{A,B}$ block. These rows correspond to equations that are consistent for any x and any y; being zero, they do not affect the rank of $R_{A,B}$.

We assume without loss of generality that columns $\mathcal{N}_A \cap \mathcal{N}_B$ are the last among the nonzero columns of B. We denote by $\breve{B}$ the matrix formed by dropping all the zero rows and columns of B and by $y_{\mathcal{N}_B}$ the vector formed by dropping elements $\mathcal{Z}_B$ from y. (For any n-vector v and a set $S \subseteq \{1, \ldots, n\}$, the notation v_S means the $|S|$-vector formed by dropping the elements of v whose indices are not in S, and similarly for matrices.) Our reduced system is

$$\breve{R}_{A,B} y_{\mathcal{N}_B} = \left[\begin{array}{ccc} & \breve{B} & \\ 0 & | & I \end{array}\right] y_{\mathcal{N}_B} = \begin{bmatrix} 0 \\ x_{\mathcal{N}_A \cap \mathcal{N}_B} \end{bmatrix},$$

where the order of the identity matrix is $|\mathcal{N}_A \cap \mathcal{N}_B|$. To test whether A is rigid with respect to B, we construct a matrix N_A whose columns span enull(A) and check

1. whether $\breve{R}_{A,B}$ has full rank, and
2. whether, for every column x in N_A,

$$\breve{R}_{A,B} \breve{R}_{A,B}^{+} \begin{bmatrix} 0 \\ x_{\mathcal{N}_A \cap \mathcal{N}_B} \end{bmatrix} = \begin{bmatrix} 0 \\ x_{\mathcal{N}_A \cap \mathcal{N}_B} \end{bmatrix},$$

where $\breve{R}_{A,B}^{+}$ denotes the Moore–Penrose pseudoinverse of $\breve{R}_{A,B}$.

If B has only a few nonzero rows and columns and if the number of columns in N_A is small, then this is an inexpensive computation. The construction is illustrated in Figure 3.1.

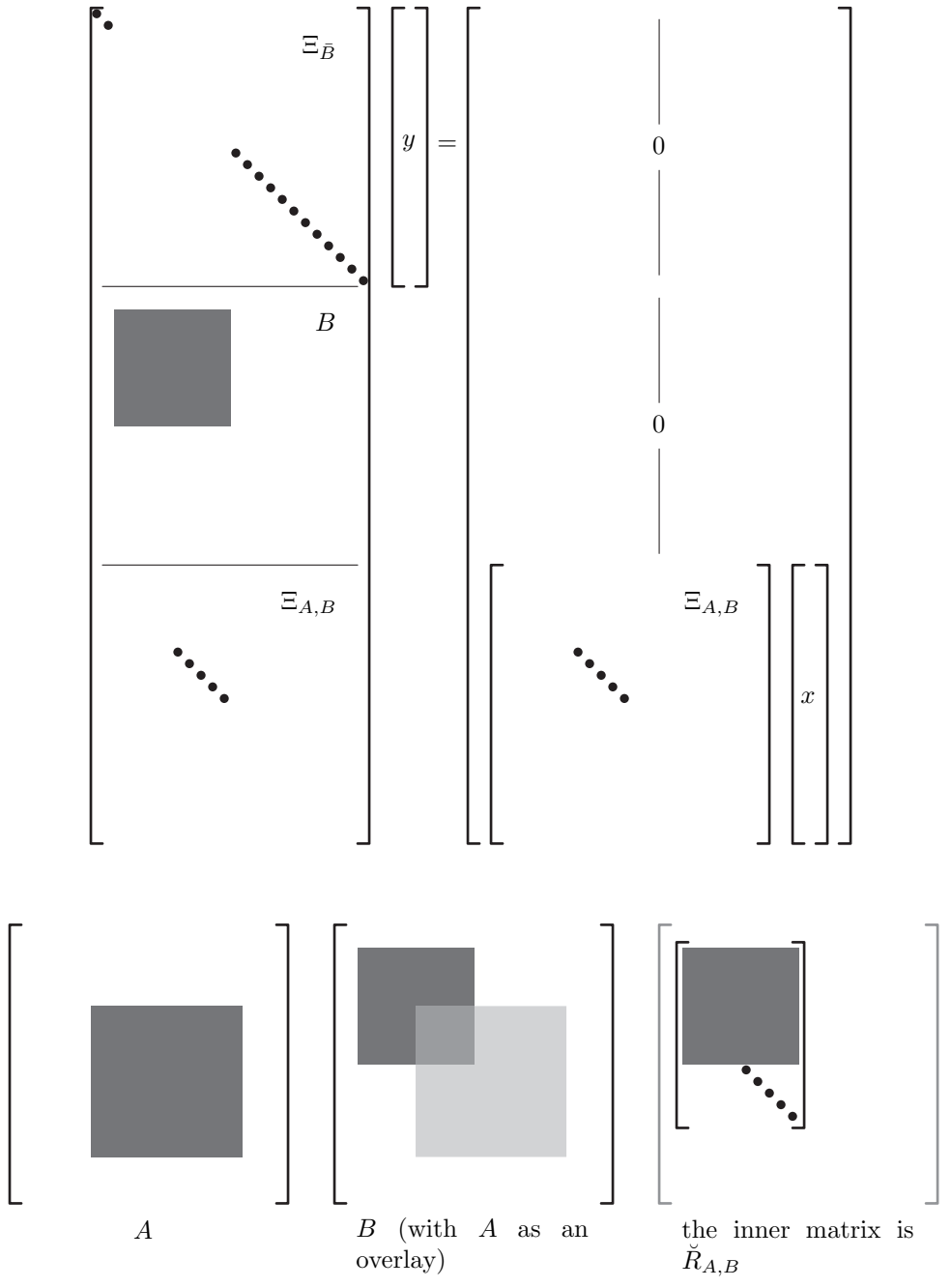


FIG. 3.1. *Testing rigidity. The top part of the figure shows the entire linear system, and the bottom part shows the construction of* $\check{R}_{A,B}$.

The next three lemmas show the relationship between null-space containment and rigidity.

LEMMA 3.4. *Let* A *be* $m \times n$ *and* B *be* $l \times n$. *Then* $\mathrm{null}(A) \subseteq \mathrm{null}(B)$ *if* A *is rigid for* B.

Proof. Let $x \in \text{enull}(A)$. Therefore, $x \in \text{null}(A)$ and $x \in \text{null}(B)$. Define $y = \Psi_B x$. We have that $x_i = y_i$ for all $i \in \mathcal{N}_A \cap \mathcal{N}_B$. By Lemma 2.4, $y \in \text{enull}(B)$. Therefore, y is a mapping of x in $\text{enull}(B)$.

We now show that y is the unique mapping of x. Let $\hat{y}$ be a mapping of x in $\text{enull}(B)$. By Lemma 2.7, $\mathcal{N}_B \subseteq \mathcal{N}_A$. The equalities $y_i = x_i = \hat{y}_i$ hold for every $i \in \mathcal{N}_A \cap \mathcal{N}_B = \mathcal{N}_B$. Therefore, $y = \hat{y}$, so y is the unique mapping of x in $\text{enull}(B)$. This implies that A is rigid with respect to B. □

LEMMA 3.5. *Let A be an $m \times n$ matrix and let B be an $l \times n$ matrix such that $\mathcal{N}_B \subseteq \mathcal{N}_A$. Then $\text{null}(A) \subseteq \text{null}(B)$*

Proof. Let $x \in \text{null}(A)$. We can write x as $y + z$, where $y \in \text{enull}(A)$ and $z \in \text{tnull}(A)$. We have that $z \in \text{tnull}(B)$, since $\mathcal{N}_B \subseteq \mathcal{N}_A$. Therefore, $z \in \text{null}(B)$.

We now show that y is also in $\text{null}(B)$. Let u be y's rigid mapping to $\text{enull}(B)$. We have that $u_i = y_i$ for every $i \in \mathcal{N}_A \cap \mathcal{N}_B = \mathcal{N}_B$. Therefore, we can write y as $y = u + u'$, where $u'_i \neq 0$ only for $i \in \mathcal{N}_A \backslash \mathcal{N}_B$. It is clear that $u' \in \text{tnull}(B) \subseteq \text{null}(B)$. Therefore, $y = u + u' \in \text{null}(B)$ and $x = y + z \in \text{null}(B)$, which is what we need to prove the lemma. □

COROLLARY 3.6. *Let A be an $m \times n$ matrix and let B be an $l \times n$ matrix such that $\mathcal{N}_A = \mathcal{N}_B$ and A and B are mutually rigid. Then $\text{null}(A) = \text{null}(B)$*

Proof. The proof directly follows from Lemmas 3.4 and 3.5. □

The last lemma in this section shows that rigidity relationships are maintained in certain Schur complements.

LEMMA 3.7. *Let A and B be $n \times n$ matrices of the form*

$$A = \begin{bmatrix} A_{11} & 0 \\ 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix},$$

where A_{11} and B_{11} are $k \times k$ matrices, $0 < k < n$, and A and B are mutually rigid. Then $\mathcal{N}_A = \{1, \ldots, k\}$ and $\mathcal{N}_B = \{1, \ldots, n\}$, and B_{22} is nonsingular. Moreover, $\text{null}(A_{11}) \subseteq \text{null}(B_{11} - B_{12}B_{22}^{-1}B_{21})$ and A_{11} is rigid with respect to $B_{11} - B_{12}B_{22}^{-1}B_{21}$

If, in addition, A and B are mutually rigid, then $\text{null}(A_{11}) = \text{null}(B_{11} - B_{12}B_{22}^{-1}B_{21})$

Proof. Let $x \in \text{null}(A_{11})$. Let $\hat{x}$ be the vector of size n that equals x in its first k coordinates and that contains zeros in its last $(n-k)$ coordinates. Clearly, $\hat{x} \in \text{null}(A)$. Since there are no zero columns in A_{11} we also have that $\hat{x} \in \text{enull}(A)$. Let $\hat{y}$ be the rigid mapping of $\hat{x}$ in $\text{enull}(B)$. The equalities $\hat{y}_i = \hat{x}_i = x_i$ hold for all $i \in \{1, \ldots, k\}$. Let y be a vector of size $(n-k)$ consisting of the last $(n-k)$ elements of $\hat{y}$. Writing the equation $B\hat{y} = 0$ in terms of x and y, we obtain

$$\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} B_{11}x + B_{12}y \\ B_{21}x + B_{22}y \end{bmatrix} = 0.$$

Multiplying the second block row by B_{22}^{-1} gives $y = -B_{22}^{-1}B_{21}x$. Substituting y with $-B_{22}^{-1}B_{21}x$ in the first block row, we get $B_{11}x - B_{12}B_{22}^{-1}B_{21}x = 0$. Therefore, $x \in \text{null}(B_{11} - B_{12}B_{22}^{-1}B_{21})$, so $\text{null}(A_{11}) \subseteq \text{null}(B_{11} - B_{12}B_{22}^{-1}B_{21})$. The containment of the null spaces, along with Lemma 3.4, shows that A_{11} is rigid with respect to $B_{11} - B_{12}B_{22}^{-1}B_{21}$.

Now assume A and B are mutually rigid (we add the assumption that B is rigid with respect to A). Let $x \in \text{null}(B_{11} - B_{12}B_{22}^{-1}B_{21})$. Let $\hat{x}$ be the vector of size n that equals x in its first k coordinates and equals $-B_{22}^{-1}B_{21}x$ in its last $(n-k)$ coordinates.

The vector $\hat{x}$ is in enull(B), since

$$B\hat{x} = \begin{bmatrix} B_{11}x + B_{12}(-B_{22}^{-1}B_{21}x) \\ B_{21}x + B_{22}(-B_{22}^{-1}B_{21}x) \end{bmatrix} = 0.$$

Because B is rigid with respect to A, the vector $\hat{x}$ has a unique mapping to enull(A). Since $\mathcal{N}_A \subseteq \mathcal{N}_B$, this mapping is $\Psi_A\hat{x}$. Therefore, $A\Psi_A\hat{x} = 0$, so $x \in$ null(A_{11}). This implies that null($B_{11} - B_{12}B_{22}^{-1}B_{21}$) $\subseteq$ null(A_{11}). Therefore, null($B_{11} - B_{12}$ $B_{22}^{-1}B_{21}$) = null(A_{11}). This concludes the proof of the lemma. ∎

4. Rigidity of sums. Finite-element matrices are sums of mostly zero matrices. This section extends our study of rigidity to sums of matrices.

LEMMA 4.1. [illegible] A [illegible] B [illegible] SPSD n [illegible] n, [illegible] $(A+B)$ [illegible] A [illegible] B

[illegible]. By Lemma 2.5, null($A + B$) $\subseteq$ null(A). Therefore, from Lemma 3.4, $(A + B)$ is rigid with respect to A. By symmetry, $(A + B)$ is rigid with respect to B, too. ∎

The previous lemma showed that a sum of SPSD matrices is rigid with respect to the terms of the sum, but the terms are not always rigid with respect to the sum, even when the terms are SPSD. For example, $A = \left[\begin{smallmatrix} 1 & -1 \\ -1 & 1 \end{smallmatrix}\right]$ is not rigid with respect to $A + \left[\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, because A is rank deficient but the sum is not. Hence, vectors in enull(A) have no mapping at all to the essential null space of the sum.

Also, the lemma holds for SPSD matrices but not for general matrices. Let $A = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ and let $B = \left[\begin{smallmatrix} 0 & -1 \\ 0 & -1 \end{smallmatrix}\right]$. Their sum is $A + B = \left[\begin{smallmatrix} 1 & -1 \\ 0 & 0 \end{smallmatrix}\right]$. The vector $\begin{bmatrix} 1 & 1 \end{bmatrix}^T \in$ enull($A + B$), but this vector has no mapping into enull(A) = $\{0\}$.

The next lemma strengthens both the hypothesis and the consequence of Lemma 4.1. It shows that if the terms are mutually rigid, then rigidity between the terms and the sum is mutual.

LEMMA 4.2. [illegible] A [illegible] B [illegible] SPSD n [illegible] n, [illegible] A [illegible] $A + B$ [illegible] B [illegible] $A + B$ [illegible]

[illegible]. By Lemma 4.1, the sum is rigid with respect to the terms. So all we need to prove is the opposite direction.

Let $x \neq 0$ be a vector in enull(A). Let y be the rigid mapping of x into enull(B). We now show that x has a mapping into enull($A + B$); we shall show the uniqueness of the mapping later. We define

$$w_i = \begin{cases} x_i, & i \in \mathcal{N}_A, \\ x_i = y_i, & i \in \mathcal{N}_A \cap \mathcal{N}_B, \\ y_i, & i \in \mathcal{N}_B, \\ 0 & \text{otherwise.} \end{cases}$$

Because $\mathcal{N}_{A+B} \subseteq \mathcal{N}_A \cup \mathcal{N}_B$, we have $w_i = 0$ for $i \in \mathcal{Z}_{A+B}$. Therefore, to show that $w \in$ enull($A + B$), we need only to show that $(A + B)w = 0$. This is indeed the case because w is an extension of both x and y, so $Aw = Bw = 0$.

We now show that w is the unique mapping of x into enull($A + B$). Suppose that there is another mapping $w' \neq w$. Under this supposition, there must be $w'_i \neq w_i$ for some $i \in \mathcal{N}_B \setminus \mathcal{N}_A$, so the restriction y' of w' to $\mathcal{N}_B$ must be different from y. By Lemmas 2.5 and 2.4, $y' \in$ enull(B). The vectors y and y' are both in enull(B)

and both coincide with x on $\mathcal{N}_A \cap \mathcal{N}_B$, so they are two different mappings of x, contradicting the hypothesis that A and B are mutually rigid. □

LEMMA 4.3. *Let A and B be SPSD $n \times n$ matrices, let C be an $m \times n$ matrix, let $A + B$ be a rigid sum with respect to C, and let $A + \alpha B$ be a rigid sum with respect to C, for some $\alpha > 0$.*

Proof. Let $\alpha > 0$. We first show that $\text{enull}(A + \alpha B) = \text{enull}(A + B)$. It is clear that $\mathcal{N}_{\alpha B} = \mathcal{N}_B$ and that αB is SPSD. From Lemma 2.6 we have $\mathcal{N}_{A+\alpha B} = \mathcal{N}_A \cup \mathcal{N}_{\alpha B} = \mathcal{N}_A \cup \mathcal{N}_B = \mathcal{N}_{A+B}$. By Lemma 2.5,

$$\text{null}(A + \alpha B) = \text{null}(A) \cap \text{null}(\alpha B) = \text{null}(A) \cap \text{null}(B) = \text{null}(A + B).$$

Therefore, $\text{enull}(A + \alpha B) = \text{enull}(A + B)$. The lemma follows directly from the definition of mutual rigidity and the fact that $\text{enull}(A + \alpha B) = \text{enull}(A + B)$. □

In some special cases, mutual rigidity between sums allows us to infer that the terms of the sums are mutually rigid and vice versa.

LEMMA 4.4. *Let A, B, and C be $n \times n$ SPSD matrices such that $\mathcal{N}_C \cap \mathcal{N}_A = \mathcal{N}_C \cap \mathcal{N}_B = \emptyset$. Let A and B be mutually rigid sums with respect to $A + C$ and $B + C$.*

Proof. Assume that A and B are mutually rigid. We show that $A + C$ is rigid with respect to $B + C$. By symmetry, $B + C$ is rigid with respect to $A + C$, so the two sums are mutually rigid.

Let $x \in \text{enull}(A + C)$. By Lemma 2.5, $x \in \text{null}(A)$ and $x \in \text{null}(C)$. Let $\hat{x}$ be x's restriction to $\mathcal{N}_A$, and let $\bar{x}$ be its restriction to $\mathcal{N}_C$. By Lemma 2.4, $\hat{x} \in \text{enull}(A)$ and $\bar{x} \in \text{enull}(C)$. Let $\hat{y}$ be $\hat{x}$'s rigid (unique) mapping to $\text{enull}(B)$. We define the vector

$$y_i = \begin{cases} \hat{y}_i, & i \in \mathcal{N}_B, \\ \bar{x}_i, & i \in \mathcal{N}_C, \\ 0 & \text{otherwise.} \end{cases}$$

The definition is valid because $\mathcal{N}_C \cap \mathcal{N}_B = \emptyset$. We show that y is the rigid mapping of x in $\text{enull}(B+C)$. Multiplying $B+C$ by y we obtain $(B+C)y = By+Cy = B\hat{y}+C\bar{x} = 0+0 = 0$. Since $y_i = 0$ for all $i \notin \mathcal{N}_B \cup \mathcal{N}_C = \mathcal{N}_{B+C}$, $y \in \text{enull}(B+C)$. By definition, $y_i = x_i$ for all $i \in (\mathcal{N}_A \cap \mathcal{N}_B) \cup \mathcal{N}_C = (\mathcal{N}_A \cup \mathcal{N}_C) \cap (\mathcal{N}_B \cup \mathcal{N}_C) = \mathcal{N}_{A+C} \cap \mathcal{N}_{B+C}$. Therefore, y is a mapping of x in $\text{enull}(B + C)$.

We now show that this mapping is indeed unique. Assume that there exists $u \in \text{enull}(B + C)$ that satisfies $u_i = x_i$ for all $i \in \mathcal{N}_{A+C} \cap \mathcal{N}_{B+C}$. We have that $u_i = x_i = y_i$ for all $i \in \mathcal{N}_C \subseteq \mathcal{N}_{A+C} \cap \mathcal{N}_{B+C}$. Let $\hat{u}$ be u's restriction to $\mathcal{N}_B$. We have $\hat{u} \in \text{enull}(B)$ and $\hat{u}_i = x_i = \hat{x}_i$ for all $i \in \mathcal{N}_A \cap \mathcal{N}_B$. Therefore, $\hat{u}$ is a mapping of $\hat{x}$ in $\text{enull}(B)$. Since A and B are mutually rigid, $\hat{u}$ must equal $\hat{y}$. Therefore, $u = y$, and y is the rigid mapping of x in $\text{enull}(B + C)$. This shows that $A + C$ is rigid with respect to $B + C$. Figure 4.1 (a) presents this notation graphically.

We now show the other direction. Assume $A + C$ and $B + C$ are mutually rigid. We show that A is rigid with respect to B; mutual rigidity follows by symmetry. The notation for this part of the proof is presented graphically in part (b) of Figure 4.1. Let $\hat{x} \in \text{enull}(A)$. Since $\mathcal{N}_C \cap \mathcal{N}_\mathcal{A} = \emptyset$, $\hat{x} \in \text{tnull}(C)$. We also have $\hat{x}_i = 0$ for all $i \notin \mathcal{N}_A \cup \mathcal{N}_C = \mathcal{N}_{A+C}$, so $\hat{x} \in \text{enull}(A+C)$. Let $\hat{y}$ be $\hat{x}$'s rigid mapping to $\text{enull}(B+C)$. We show that $\hat{y}$ is $\hat{x}$'s rigid mapping to $\text{enull}(B)$. By Lemma 2.5, $\hat{y} \in \text{null}(B)$. Also, $\hat{y}_i = \hat{x}_i = 0$ for all i in $i \in \mathcal{N}_C \subseteq \mathcal{N}_{B+C}$. Therefore, $\hat{y}_i = 0$ for all $i \notin \mathcal{N}_B$, so $\hat{y} \in \text{enull}(B)$. By definition, $\hat{x}_i = \hat{y}_i$ for all $(\mathcal{N}_A \cap \mathcal{N}_B) \subseteq \mathcal{N}_{A+C} \cap \mathcal{N}_{B+C}$. This implies that $\hat{y}$ is a mapping of $\hat{x}$ in $\text{enull}(B)$.

Finally, we claim that $\hat{y}$ is the unique mapping of $\hat{x}$. Assume that there exists $\hat{u}$ in enull(B) that satisfies $\hat{x}_i = \hat{u}_i$ for all $i \in \mathcal{N}_A \cap \mathcal{N}_B$. We have that $\hat{x}_i = \hat{u}_i = 0$ for all $i \in \mathcal{N}_C$. Since $\hat{u}$ is also in enull($B+C$), it is the rigid mapping of $\hat{x} \in$ enull($A+C$) in enull($B+C$). Because $A+B$ is rigid with respect to $A+C$, we have that $\hat{u} = \hat{y}$. Therefore, $\hat{y}$ is indeed unique. This implies that A is rigid with respect to B, which concludes the proof of the lemma. ∎

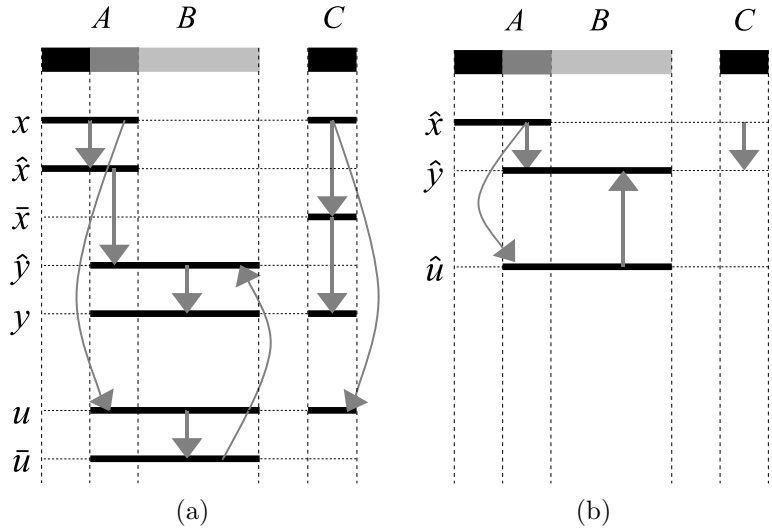


FIG. 4.1. *An illustration of the notation of Lemma* 4.4: (a) *The vectors defined in the proof of the mutual rigidity of* $A+B$ *and* $A+C$. (b) *The vectors defined in the proof of the mutual rigidity of* A *and* B.

We would like to build larger assemblies of mutually rigid matrices from chains of mutual rigidity, but this is not always possible, as the next example shows.

4.5. Let

$$A = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

These matrices are all SPSD, and their essential null spaces are spanned by $\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T$, $\begin{bmatrix} 0 & 1 & 1 \end{bmatrix}^T$, and $\begin{bmatrix} 1 & 0 & -1 \end{bmatrix}^T$, respectively. The matrices A and B are mutually rigid, and so are B and C. The essential null space of $A+B$ is spanned by $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$, and the essential null space of $B+C$ is spanned by $\begin{bmatrix} 1 & -1 & -1 \end{bmatrix}^T$. Therefore, C is not mutually rigid with $A+B$ and A is not mutually rigid with $B+C$. Moreover, none of A, B, C, $A+B$, or $B+C$ is mutually rigid with $A+B+C$, because $A+B+C$ has full rank. This example is inspired by the analysis of signed graphs in [4], which shows that $A+B+C$ has full rank.

To build larger assemblies of mutually rigid matrices, we need another tool.

5. Null-space compatibility. This section defines and explores a concept that we call null-space compatibility, which is the tool that allows us to build large assemblies of mutually rigid matrices.

DEFINITION 5.1. [illegible] $\mathbb{S} \subseteq \mathbb{R}^n$ [illegible] A [illegible] $\mathbb{S}$-compatible ([illegible] $\mathbb{S}$) [illegible] enull(A) [illegible] $\mathbb{S}$ [illegible] $\mathbb{S}$ [illegible] $\mathcal{N}_A$ [illegible] enull(A)

DEFINITION 5.2. [illegible] A [illegible] n [illegible] A $\mathbb{S}$ [illegible] $\mathcal{N}_A = \{1, \ldots, n\}$ [illegible] A $\mathbb{S}$ rigid [illegible] null$(A) = \mathbb{S}$ [illegible] $\mathcal{N}_A = \{1, \ldots, n\}$ $\mathbb{S}$ [illegible] A [illegible]

Given a basis for $\mathbb{S}$, we can easily check the compatibility of a matrix A. Let the columns of N be a basis for $\mathbb{S}$, and let N_A be a basis for enull(A). A is compatible with $\mathbb{S}$ if and only if $N_A = \Psi_A N_A$ and $\Psi_A N$ have the same range. This can be checked numerically using the singular value decompositions of the two matrices, for example.

[illegible] 5.3. Let $\mathbb{S} = \text{span}\begin{bmatrix}1 & 1 & 1\end{bmatrix}^T$. The matrices A and B from Example 4.5 are compatible with $\mathbb{S}$, but C is not. Note that the mutual rigidity of A and C together with the $\mathbb{S}$-compatibility of A do not imply $\mathbb{S}$-compatibility for C. The matrix $A + B$ from the same example is also compatible with $\mathbb{S}$, and since $\mathcal{N}_{A+B} = \{1, 2, 3\}$, $A + B$ is rigid.

LEMMA 5.4. [illegible] A [illegible] $\mathbb{S}$ [illegible] N [illegible] $\mathbb{S}$ [illegible] enull$(A) = \text{span}(\Psi_A N)$

[illegible]. We first show that enull$(A) \subseteq \text{span}(\Psi_A N)$. Let $x \in \text{enull}(A)$. Since A is $\mathbb{S}$-compatible, x has a unique extension w in $\mathbb{S}$. By definition, there exists a vector y such that $w = Ny$. Substituting w in the equation $x = \Psi_A w$, we get $x = \Psi_A N y$. Therefore, $x \in \text{span}(\Psi_A N)$, so enull$(A) \subseteq \text{span}(\Psi_A N)$.

We now show that $\text{span}(\Psi_A N) \subseteq \text{enull}(A)$. Let $x = \Psi_A N y \in \text{span}(\Psi_A N)$. Define $w = Ny \in \mathbb{S}$. Since A is $\mathbb{S}$-compatible, $x = \Psi_A w \in \text{enull}(A)$. This shows that $\text{span}(\Psi_A N) \subseteq \text{enull}(A)$. ∎

The definition of null-space compatibility is related to the definition of mutual rigidity, but it defines compatibility with respect to a space, not with respect to a particular matrix having that space as a null space. Here is the relationship of $\mathbb{S}$-compatibility with mutual rigidity.

LEMMA 5.5. [illegible] $\mathbb{S} \subseteq \mathbb{R}^m$ [illegible] B [illegible] $\mathbb{S}$ [illegible] A $\mathbb{S}$ [illegible] A [illegible] B [illegible]

[illegible]. The equivalence follows from the fact that enull$(B) = \text{null}(B) = \mathbb{S}$ (because $\mathcal{Z}_B = \emptyset$) and from the fact that $\mathcal{N}_A \cap \mathcal{N}_B = \mathcal{N}_A$. ∎

If the dimension of $\mathbb{S}$ is small, the $\mathbb{S}$-compatibility test given after Definition 5.2 can be much more efficient than the test for mutual rigidity given earlier.

[illegible] 5.6. Two matrices that are both compatible with some null space $\mathbb{S}$ are not necessarily mutually rigid. For example,

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

are both compatible with $\mathbb{S} = \text{span}\begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix}^T$, but they are not mutually rigid. Also, their sum is not $\mathbb{S}$-compatible. Since $\mathcal{N}_{A+B} = \{1, 2, 3, 4\}$, enull$(A + B) =$

$\text{null}(A+B)$, and so $A+B$ is $\mathbb{S}$-compatible if and only if $\text{null}(A+B) = \mathbb{S}$. However,

$$\text{enull}(A+B) = \text{enull}\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} = \text{span}\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \neq \mathbb{S} = \text{span}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

The next two lemmas are key results that will allow us to build large assemblies of mutually rigid matrices.

LEMMA 5.7. *A and B are mutually rigid with respect to $\mathbb{S}$ if and only if $\mathbb{S}$ is the only subspace of $\text{enull}(A+B)$ that is also in $\mathbb{S}$.*

Proof. Let u be a vector in $\text{enull}(A+B)$, let x be the restriction of u to $\mathcal{N}_A$, and let y be the restriction to $\mathcal{N}_B$. Clearly, y is the rigid mapping of x in $\text{enull}(B)$. Let w be the extension of x to a vector in $\mathbb{S}$. We claim that w is a unique extension of u to $\mathbb{S}$. If w is not an extension of u, then they must differ on an index in $\mathcal{N}_B \setminus \mathcal{N}_A$, so the restriction of w to $\mathcal{N}_B$ is some $y' \neq y$. The vector y' is also a mapping of x in $\text{enull}(B)$. But both y and $y' \neq y$ are mappings of x in $\text{enull}(B)$, in contradiction to the mutual rigidity of A and B.

We now show that w is the unique extension of u to $\mathbb{S}$. If there is another extension, its restriction to $\mathcal{N}_A$ must differ from x, so it cannot be an extension of u.

We now show that the restriction u of a vector $w \in \mathbb{S}$ is in $\text{enull}(A+B)$. The restriction x of u to $\mathcal{N}_A$ is also the restriction of w to $\mathcal{N}_A$, so $Ax = Au = Aw = 0$. The same is true for the restrictions to $\mathcal{N}_B$. Therefore, $(A+B)u = 0$, so $u \in \text{enull}(A+B)$. ∎

We now introduce a technical lemma that shows how to transform a null-space extension of a vector into the rigid mapping of the same vector.

LEMMA 5.8. *A and B are mutually rigid with respect to $\mathbb{S}$. $x \in \text{enull}(A)$, $w \in \mathbb{S}$, and $\Psi_B w$ is the extension of $x \in \text{enull}(B)$ to u. $x \in \text{enull}(B)$ and $u \in \mathbb{S}$, w is unique.*

Proof. Let $y = \Psi_B w$. We first show that y is the rigid mapping of x to $\text{enull}(B)$. The vector y is in $\text{enull}(B)$, since B is $\mathbb{S}$-compatible and $w \in \mathbb{S}$. From the definition of w and y we have that $x_i = w_i = y_i$ for all $i \in \mathcal{N}_A \cap \mathcal{N}_B$. Therefore, y is a mapping of x in $\text{enull}(B)$, and it is unique because A and B are mutually rigid.

Let u be the rigid mapping of x to $\text{enull}(B)$. Since this mapping is unique, $u = y = \Psi_B w$. The vector w is an extension of u to $\mathbb{S}$. The matrix B is $\mathbb{S}$-compatible, so this extension is unique. ∎

The following lemma is the main result of this section. Compare this lemma to Example 4.5: in the example, the three matrices were not all compatible with some null space $\mathbb{S}$; the conclusion of this lemma does not hold in that example.

LEMMA 5.9. *A, B, C SPSD $n \times n$. $\mathbb{S}$ is a subspace of $\mathbb{R}^n$. A and B are mutually rigid with respect to $\mathbb{S}$, and B and C are mutually rigid with respect to $\mathbb{S}$. $A+B$ and C are mutually rigid with respect to ($B+C$ and A).*

Proof. We first show that $A+B$ is rigid with respect to C. Let u be in $\text{enull}(A+B)$. By Lemma 5.7, the matrix $A+B$ is $\mathbb{S}$-compatible. Let w be u's extension to $\mathbb{S}$. Define $x = \Psi_A^T w$, $y = \Psi_B^T w$, and $z = \Psi_C^T w$. By definition, $u = \Psi_{A+B}^T w$ and therefore $z_i = u_i$ for all $i \in \mathcal{N}_{A+B} \cap \mathcal{N}_C$. Since C is $\mathbb{S}$ compatible, $z \in \text{enull}(C)$. Therefore, z is a mapping of u in $\text{enull}(C)$.

We show that z is unique. The matrices $A+B$ and A are mutually rigid according to Lemma 4.2. According to Lemma 5.8, x is the unique mapping of u in $\text{enull}(A)$, y

is the unique mapping of x in enull(B), and z is the unique mapping of y in enull(C). Therefore, z is the unique mapping of u in enull(C) and $A + B$ is rigid with respect to C.

We show now that C is rigid with respect to $A + B$. Let z be in enull(C). The matrix C is $\mathbb{S}$-compatible. Let w be z's extension to $\mathbb{S}$. Define $x = \Psi_A^T w$, $y = \Psi_B^T w$, and $u = \Psi_{A+B}^T w$. By definition, $z = \Psi_C^T w$ and therefore $z_i = u_i$ for all $i \in \mathcal{N}_{A+B} \cap \mathcal{N}_C$. Therefore, u is a mapping of z in enull($A + B$).

We show that u is unique. According to Lemma 5.8, y is the unique mapping of z in enull(B), x is the unique mapping of y in enull(A), and u is the unique mapping of x in enull($A + B$). Therefore, u is the unique mapping of z in enull($A + B$). This implies that C is rigid with respect to $A + B$ and concludes the proof of the lemma. □

5.10. This example shows that A and C are not necessarily mutually rigid, even if A and B are mutually rigid, B and C are mutually rigid, and all three are $\mathbb{S}$-compatible. Instead of constructing the matrices here, we refer the reader to the construction of section 7.3, where we construct matrices that model elastic triangles. Suppose that the three matrices are 10-by-10, that A corresponds to a triangle (p_1, p_2, p_3) (p_i is a point in the plane; all the points are different), B to $(p_2, , p_3, p_4)$, and C to (p_3, p_4, p_5). We assume that the three triangles are disjoint except for shared edges. The analysis of section 7.3 shows that A and B are mutually rigid and B and C are mutually rigid but A and C are not.

The next lemma characterizes the rigidity and $\mathbb{S}$-compatibility of certain larger sums.

LEMMA 5.11. [illegible] A [illegible] $B_1, B_2, \ldots, B_k$ [illegible] SPSD [illegible] $\mathbb{S}$ [illegible] A [illegible] B_i [illegible] $i = 1, \ldots, k$ [illegible] A [illegible] $A + \sum_{i=1}^{k} B_i$ [illegible] $A + \sum_{i=1}^{k} B_i$ [illegible] $\mathbb{S}$ [illegible].

[illegible]. We prove the lemma by induction on k. The case $k = 1$ is trivial by Lemmas 4.2 and 5.7. We assume that the claim is correct for k smaller than n and show that it is correct for $k = n$. By the inductive assumption, A and $A + \sum_{i=1}^{n-1} B_i$ are mutually rigid and $A + \sum_{i=1}^{n-1} B_i$ is $\mathbb{S}$-compatible. A and B_n are mutually rigid. Therefore, by Lemma 5.9 we have that B_n and $A + A + \sum_{i=1}^{n-1} B_i = 2A + \sum_{i=1}^{n-1} B_i$ are mutually rigid. By Lemma 4.3, B_n and $A + \sum_{i=1}^{n-1} B_i$ are mutually rigid. Therefore, by Lemma 5.9, we have that A and $B_n + A + \sum_{i=1}^{n-1} B_i = A + \sum_{i=1}^{n} B_i$ are mutually rigid. By Lemma 5.7, we also have that $B_n + A + \sum_{i=1}^{n-1} B_i = A + \sum_{i=1}^{n} B_i$ is $\mathbb{S}$-compatible. This concludes the proof of the lemma. □

The following lemma shows that the property of null-space compatiblity simplifies considerably the mutual rigidity test.

LEMMA 5.12. [illegible] A [illegible] B [illegible] $\mathbb{S}$ [illegible] SPSD n [illegible] n [illegible] N [illegible] n [illegible] ℓ [illegible] ℓ [illegible] $\mathbb{S}$ [illegible] A [illegible] B [illegible] $\Psi_{A,B} N$ [illegible] ℓ

[illegible]. We first assume that $\Psi_{A,B} N$ has rank ℓ and show that A is rigid with respect to B. Mutual rigidity follows by symmetry. Let $x \in$ enull(A). We can write $x = \Psi_A N y_A$, where y_A is some length-ℓ vector. Define $u = \Psi_B N y_A$. By definition, $u \in$ enull(B) and since

$$\Psi_{A,B} u = \Psi_{A,B} \Psi_B N y_A = \Psi_{A,B} N y_A = \Psi_{A,B} \Psi_A N y_A = \Psi_{A,B} x,$$

u is a mapping of x.

We now show it is unique. Let $\hat{u} \in$ enull(B), such that $\Psi_{A,B} \hat{u} = \Psi_{A,B} x$. By

definition, $\hat{u} = \Psi_B N y_B$ for some y_B. Clearly,

$$\Psi_{A,B}\hat{u} = \Psi_{A,B}\Psi_B N y_B = \Psi_{A,B} N y_B.$$

Therefore,

$$\Psi_{A,B} N y_A = \Psi_{A,B} x = \Psi_{A,B}\hat{u} = \Psi_{A,B} N y_B,$$

and

$$\Psi_{A,B} N (y_A - y_B) = 0.$$

Since $\Psi_{A,B}N$ has full rank ℓ, $y_A = y_B$, so $u = \hat{u}$. This implies that A is rigid with respect to B.

We show now the other direction. Assume A and B are mutually rigid. We note that for every $0 \neq n \in \mathbb{S}$, $\Psi_A n \neq 0$. Otherwise, both n and 0 are extensions of $0 \in \text{enull}(A)$, which is impossible since A is $\mathbb{S}$-compatible.

Suppose for contradiction that $\Psi_{A,B}N$ is rank deficient. Therefore, there exists an ℓ-vector $y \neq 0$ such that $\Psi_{A,B} N y = 0$. Since N has full rank, $Ny \neq 0$. By the previous paragraph, $\Psi_A N y \neq 0$. Note that the mapping of $\Psi_A N y$ to $\text{enull}(B)$ is $\Psi_B \Psi_A N y = \Psi_{A,B} N y = 0$. Therefore, the mapping of both the zero vector and of $\Psi_A N y$ to $\text{enull}(B)$ is the zero vector. This contradicts the assumption that A and B are mutually rigid. ∎

6. The rigidity graph. Mutual rigidity relationships in a collection of $\mathbb{S}$-compatible SPSD matrices define a graph structure that we can use to demonstrate the rigidity of finite-element matrices.

DEFINITION 6.1. [illegible] $A_1, A_2, \ldots, A_k$ [illegible] $\mathbb{S}$ [illegible] SPSD $n \times n$ [illegible] $\mathbb{S}$ [illegible] rigidity graph $G = (V, E)$ [illegible] $\{A_1, \ldots, A_k\}$ [illegible] $V = \{A_1, \ldots, A_k\}$ [illegible]

$$E = \{(A_e, A_f) : A_e \text{ [illegible] } A_f \text{ [illegible]}\}.$$

We could also define the rigidity graph of a collection of matrices that are not necessarily $\mathbb{S}$-compatible, but Example 4.5 suggests that such a definition might not have interesting applications. On the other hand, the $\mathbb{S}$-compatibility requirement in the definition enables an important result, which we state and prove next.

LEMMA 6.2. [illegible] G [illegible] $A_1, A_2, \ldots, A_k$ [illegible] $\mathbb{S}$ [illegible] SPSD $n \times n$ [illegible] H [illegible] G [illegible] $V(H)$ [illegible] H [illegible] A_e [illegible] H [illegible] A_e [illegible] $\sum_{A_f \in V(H)} A_f$ [illegible] $\sum_{A_f \in V(H)} A_f$ [illegible] $\mathbb{S}$ [illegible]

[illegible]. Let T be a depth-first-search tree of H whose root is A_e. Denote by $\{T_1, T_2, \ldots, T_c\}$ the trees in the forest formed from T by removing A_e. We show by induction on the height h of T that the following claims holds: A_e and $A_e + \sum_{i=1}^{c} \sum_{A_f \in T_i} A_f$ are mutually rigid, and $A_e + \sum_{i=1}^{c} \sum_{A_f \in T_i} A_f$ is $\mathbb{S}$-compatible.

The claim holds trivially for $k = 1$ (a single-vertex tree), because A_e is $\mathbb{S}$-compatible and is mutually rigid with itself.

Now, we assume that the inductive claim is correct for trees with height h or less and we show it is correct for trees with height $h+1$. Let T be a tree of height $h+1$ whose root vertex is A_e, and let $T_1, T_2, \ldots, T_c$ be the subtrees defined above. The height of every T_i is h or less. Let A_i be the root vertex of T_i, and let F_i be the forest of A_i's descendants. By definition, A_e and A_i are mutually rigid. By the inductive claim on

T_i, we have that $A_i + \sum_{A_f \in F_i} A_f$ is $\mathbb{S}$-compatible and mutually rigid with A_i. We note that all the sums of the form $\sum A_f$ are symmetric and positive semidefinite. Therefore, by Lemma 5.9 A_e and $A_i + (A_i + \sum_{A_f \in F_i} A_f) = 2A_i + \sum_{A_f \in F_i} A_f$ are mutually rigid. By Lemma 4.3, we have that A_e and $A_i + \sum_{A_f \in F_i} A_f = \sum_{A_f \in T_i} A_f$ are mutually rigid for every i. By Lemma 5.11, we have that A_e and $A_e + \sum_{i=1}^{c} \sum_{A_f \in T_i} A_f$ are mutually rigid and $A_e + \sum_{i=1}^{c} \sum_{A_f \in T_i} A_f$ is $\mathbb{S}$-compatible. This concludes the proof of the lemma. ∎

The next result generalizes the previous lemma.

THEOREM 6.3. [illegible] G [illegible] $A_1, A_2, \ldots, A_k$ [illegible] $\mathbb{S}$ [illegible] SPSD n [illegible] n [illegible] H_1 [illegible] H_2 [illegible] G [illegible] A_e [illegible] $B = \sum_{A_f \in V(H_1)} A_f$ [illegible] $C = \sum_{A_f \in V(H_2)} A_f$ [illegible]

[illegible]. According to Lemma 6.2, B and A_e are mutually rigid, and so are C and A_e. By Lemma 5.9, we have that B and $A_e + C$ are mutually rigid. The sum $A_e + C$ equals $\sum_{A_f \in V(H_2) \setminus A_e} A_f + 2A_e$. By Lemma 4.3, we have that B and $\sum_{A_f \in V(H_2)} A_f = C$ are mutually rigid. ∎

The next theorem shows that the rigidity graph can sometimes tell us that a finite-element matrix is rigid in the sense that its null space is exactly $\mathbb{S}$. This is only a sufficient condition; it is not necessary.

THEOREM 6.4. [illegible] G [illegible] $A_1, A_2, \ldots, A_k$ [illegible] $\mathbb{S}$ [illegible] n [illegible] n [illegible] $A = \sum_{e=1}^{k} A_e$ [illegible] N [illegible] $\mathbb{S}$ [illegible] G [illegible] $\mathrm{enull}(A) = \mathrm{span}(\Psi_A N)$ [illegible] $\bigcup_{e=1}^{k} \mathcal{N}_{A_e} = \{1, \ldots, n\}$ [illegible] $\mathrm{null}(A) = \mathbb{S}$ [illegible] A [illegible]

[illegible]. According to Lemma 6.2, A is $\mathbb{S}$-compatible. Therefore, by Lemma 5.4, $\mathrm{enull}(A) = \mathrm{span}(\Psi_A N)$.

If $\bigcup_{e=1}^{k} \mathcal{N}_{A_e} = \{1, \ldots, n\}$, then $\Psi_A N = N$ and $\mathrm{enull}(A) = \mathrm{null}(A)$. Therefore, $\mathrm{null}(A) = \mathrm{enull}(A) = \mathrm{span}(\Psi_A N) = \mathrm{span}(N) = \mathbb{S}$. By definition, A is rigid. ∎

When the rigidity graph is not connected, the null space may or may not be $\mathbb{S}$. To show that a disconnected rigidity graph sometimes corresponds to a null space larger than $\mathbb{S}$, consider

$$A_1 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}.$$

Both are compatible with $\mathbb{S} = \mathrm{span}\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$, but they are not mutually rigid. Therefore, their rigidity graph consists of two disconnected vertices. The null space of $A_1 + A_2$ is spanned by both $\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$ and $\begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}^T$, so it is indeed larger than $\mathbb{S}$, even though $\mathcal{N}_{A_1} \cup \mathcal{N}_{A_2} = \{1, 2, 3, 4\}$. Examples in which the rigidity graph is not connected but the null space of the sum is $\mathbb{S}$ are more complex; we show an example in section 7.3.

7. Three case studies. This section presents three families of $\mathbb{S}$-compatible SPSD matrices for two different $\mathbb{S}$s. One is well known and we present it without proofs. The second and third are more complex and we present them in full.

7.1. Laplacians. The first family of matrices that we present consists of Laplacians, matrices that are often used in spectral graph theory and in other areas. The results in this section are all adaptations of known results, so we omit the proofs. All the matrices and vectors are of order n.

DEFINITION 7.1. [illegible] (k,j) [illegible] $1 \le k < j \le n$ [illegible] $u^{(k,j)}$

$$u_i^{(k,j)} = \begin{cases} +1, & i = k, \\ -1, & i = j, \\ 0 & \text{[illegible]} \end{cases}$$

[illegible] (k,j) edge matrix [illegible] $A^{(k,j)} = u^{(k,j)}\, u^{(k,j)\,T}$

LEMMA 7.2. [illegible] $1 \le k < j \le n$ [illegible] $A^{(k,j)}$ [illegible]

1. $A^{(k,j)}$ [illegible] SPSD
2. $\mathcal{N}_{A^{(k,j)}} = \{k, j\}$
3. $A^{(k,j)}$ [illegible] $\mathbb{S}_1 = \operatorname{span}\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}^T$

The next lemma gives a sufficient and necessary condition for two edge matrices to be mutually rigid.

LEMMA 7.3. [illegible] $A^{(i,j)}$ [illegible] $A^{(k,\ell)}$ [illegible] $|\{i,j\} \cap \{k,\ell\}| \ge 1$

Laplacians are sums of edge matrices (sometimes of positively scaled edge matrices). They are often defined using an undirected graph $G = (\{1, 2, \ldots, n\}, E)$,

$$A^{(G)} = \sum_{(i,j)\in E} A^{(i,j)}.$$

Each edge matrix $A^{(i,j)}$ is then associated with an edge $(i,j) \in E$ in the graph G. Lemma 7.3 states that two edge matrices are mutually rigid if and only if the corresponding edges are incident on a common vertex.

The rigidity graph of $\{A^{(i,j)} | (i,j) \in E\}$ is a dual of G:

$$G_{\text{dual}} = \left(E, \{(e,f) : e \text{ and } f \text{ share a vertex in } G\}\right).$$

The rigidity graph of Laplacians is special in that its connectivity is not only a sufficient condition for the rigidity of the Laplacian, as shown in Theorem 6.4, but also a necessary condition.

LEMMA 7.4. [illegible] $G = (\{1, 2, \ldots, n\}, E)$ [illegible] $A^{(G)} = \sum_E A^{(i,j)}$ [illegible] G_{dual} [illegible] $\{A^{(i,j)} | (i,j) \in E\}$ [illegible]

[illegible]. We first show that if $A^{(G)}$ is rigid, then G is a connected graph. Assume for contradiction that G is not connected. Therefore, there are two nonempty sets of vertices S and $\bar{S} = \{1, \ldots, n\} \backslash S$ that are disconnected. Define the vector x:

$$x_i = \begin{cases} +1, & i \in S, \\ -1, & i \in \bar{S}. \end{cases}$$

By definition, $A^{(i,j)}x = 0$ for every $(i,j) \in E$. Therefore, $A^{(G)}x = \sum_{(i,j)\in E} A^{(i,j)}x = 0$. The vector x is in enull$(A^{(G)})$ and has no extension to $\mathbb{S}_1$. This contradicts the

assumption that $A^{(G)}$ is rigid, since this assumption implies that it is $\mathbb{S}_1$-compatible. Therefore, G is a connected graph.

It is clear that if G is connected, then G_{dual} is connected. This concludes the proof of the lemma. □

All the results on the rigidity of Laplacians hold for weighted Laplacians, which are sums of positively scaled edge matrices.

7.2. Elastic struts in two dimensions. The second family of matrices model a collection of pin-jointed struts. Such a collection may form a rigid structure called a truss (e.g., a triangle in two dimensions) or a nonrigid structure called a mechanism (e.g., two struts connected by a pin). The rigidity graph of such a structure, however, is never connected: it has no edges at all. Therefore, the rigidity graph can never show that the underlying structure is rigid. This shows that our analysis indeed only provides sufficient, but not necessary, conditions. We carry out the analysis nonetheless to give another example of how to derive mutual rigidity conditions.

We note that there is a combinatorial technique that can determine whether such a structure is rigid, under a technical assumption on the geometrical location of the pins. The structure is modeled by a graph in which vertices correspond to pins (assuming there is a pin at the end of each strut) and in which edges correspond to struts. If the pins are in an appropriately defined general position, then several equivalent conditions on the graph characterize exactly the rigidity of the structure [13, 15, 17, 21, 14]. These conditions can be tested in $O(n^2)$ operations [12].

Our technique is more general but less precise than these techniques. It applies to any finite-element matrix, but it provides only sufficient conditions for rigidity. In the cases of two-dimensional struts, our sufficient conditions are never satisfied. We show later in this section that our technique does work for other families of elastic structures.

DEFINITION 7.5. [illegible] $P = \{p_l\}_{l=1}^{n}$ [illegible] $p_l = (x_l, y_l)$ [illegible] $p_i \neq p_j$ [illegible] $v^{(i,j)}$ [illegible] $2P$ [illegible]

$$v_i^{(k,j)} = \begin{cases} (x_k - x_j)/r_{k,j}, & i = 2k-1, \\ (y_k - y_j)/r_{k,j}, & i = 2k, \\ -(x_k - x_j)/r_{k,j}, & i = 2j-1, \\ -(y_k - y_j)/r_{k,j}, & i = 2j, \\ 0 & \text{[illegible]}, \end{cases}$$

[illegible] $r_{k,j} = \sqrt{(x_k - x_j)^2 + (y_k - y_j)^2}$ [illegible] (i,j) strut matrix [illegible] $A^{(i,j)} = v^{(i,j)}\, v^{(i,j)\,T}$

DEFINITION 7.6. [illegible] $P = \{p_l\}_{l=1}^{n}$ [illegible]

$$N_P^{(x)} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ \\ \\ 1 \\ 0 \end{bmatrix}, \quad N_P^{(y)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ \\ \\ 0 \\ 1 \end{bmatrix}, \quad \text{[illegible]} \quad N_P^{(r)} = \begin{bmatrix} -y_1 \\ x_1 \\ -y_2 \\ x_2 \\ \\ \\ -y_n \\ x_n \end{bmatrix}.$$

planar null space $\mathbb{S}_P = \operatorname{span} N_P$ $N_P = [N_P^{(x)}\ N_P^{(y)}\ N_P^{(r)}]$

The next lemma shows that the strut matrices are $\mathbb{S}_P$ compatible.

LEMMA 7.7. $P = \{p_l\}_{l=1}^n$

x y

1. $A^{(i,j)}$
2. $\mathcal{N}_{A^{(i,j)}} = \{2i-1, 2i, 2j-1, 2j\}$
3. $A^{(i,j)}$ $\mathbb{S}_P$

The first two claims in the lemma follow directly from the definition of $A^{(i,j)}$. We show that $A^{(i,j)}$ is $\mathbb{S}_P$-compatible by showing that the columns of $\Psi_{A^{(i,j)}} N_P$ form a basis for enull$(A^{(i,j)})$.

A direct calculation, which we omit, shows that $A\Psi_{A^{(i,j)}} N_P = 0$. The points p_i and p_j are different, so the rank of $\Psi_{A^{(i,j)}} N_P$ is 3. The rank of $A^{(i,j)}$ is 1, so its essential null space has dimension 3. Therefore, $\Psi_{A^{(i,j)}} N_P$ spans enull$(A^{(i,j)})$, so $A^{(i,j)}$ is $\mathbb{S}_P$ compatible. ∎

The following lemma indicates that the rigidity graph of a collection of strut matrices contains only trivial edges (self loops, which are always present).

LEMMA 7.8. $P = \{p_l\}_{l=1}^n$

x y $A^{(i,j)}$ $A^{(k,\ell)}$

$A^{(i,j)}$ $A^{(k,\ell)}$ $\{i,j\} = \{k,\ell\}$

Assume $A^{(i,j)}$ and $A^{(k,\ell)}$ are mutually rigid. By Lemma 5.12, rank $(\Psi_{A^{(i,j)},A^{(k,\ell)}} N_P) = 3$. Therefore, using Lemma 7.7, $|\{2i-1, 2i, 2j-1, 2j\} \cap \{2k-1, 2k, 2\ell-1, 2\ell\}| \geq 3$. This implies that $\{i,j\} = \{k,\ell\}$. The other direction is immediate; a matrix is always mutually rigid with itself. ∎

7.3. Elastic triangles in two dimensions. We now study another family of matrices that also arises in two-dimensional linear elasticity—matrices that model triangular elements. The rigidity graph of such a collection can be connected, so the rigidity graph can sometimes tell us that the structure is rigid. There are also cases in which the structure is rigid but its rigidity graph is not connected.

DEFINITION 7.9. $P = \{p_l\}_{l=1}^n$ $p_l = (x_l, y_l)$ $v^{(i,j)}$ $A^{(i,j)}$ 7.5

p_i p_j p_k (i,j,k) element matrix

$$A^{(i,j,k)} = A^{(i,j)} + A^{(i,k)} + A^{(j,k)} = v^{(i,j)}\, v^{(i,j)\,T} + v^{(j,k)}\, v^{(j,k)\,T} + v^{(k,i)}\, v^{(k,i)\,T}.$$

$A^{(i,j,k)}$

)

The next lemma is the equivalent of Lemma 7.7. We omit the proof, which is similar to the proof of Lemma 7.7.

LEMMA 7.10. $P = \{p_l\}_{l=1}^n$

A (i,j,k)

1. A
2. $\mathcal{N}_A = \{2i-1, 2i, 2j-1, 2j, 2k-1, 2k\}$
3. A $\mathbb{S}_P$ $\mathbb{S}_P$ 7.6.

The following lemma characterizes mutual rigidity between $\mathbb{S}_P$-compatible matrices.

LEMMA 7.11. $P = \{p_l\}_{l=1}^n$ A

B $\mathbb{S}_P$ i $j \neq i$

$\{2i-1, 2i, 2j-1, 2j\} \subseteq \mathcal{N}_A \cap \mathcal{N}_B$ A B

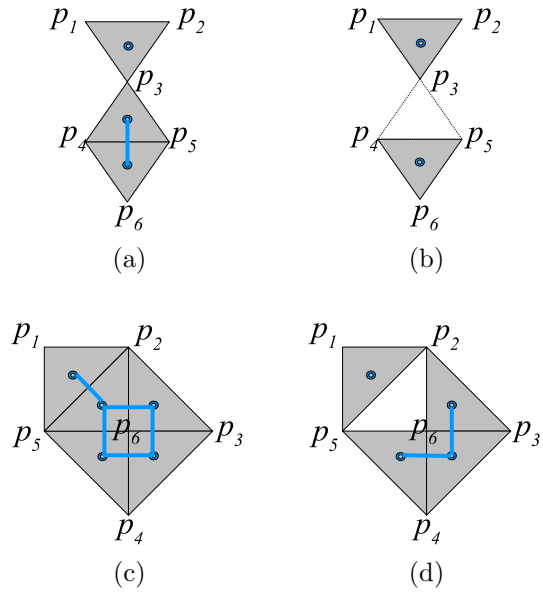


FIG. 7.1. *Triangular plane elements and their rigidity graphs (in blue).*

. By Lemma 5.12, it is sufficient to show that $\Psi_{A,B}N_P$ is rank-3. Indeed, it is easy to validate that since the points are different, $\text{rank}(\Psi_{A,B}N_P) > 3$. ∎

The last lemma of this section shows how to construct the rigidity graph for this family of matrices.

LEMMA 7.12. . $P = \{p_l\}_{l=1}^n$ $A = A^{(i,j,k)}$ $B = A^{(p,q,r)}$ A B $|\{i,j,k\} \cap \{p,q,r\}| \geq 2$

. If $|\{i,j,k\} \cap \{p,q,r\}| \geq 2$, there exist $m \neq l$ such that $\{2m-1, 2m, 2l-1, 2l\} \subseteq \mathcal{N}_A \cap \mathcal{N}_B$. Then, by Lemma 7.11, A and B are mutually rigid.

If A and B are mutually rigid, by Lemma 5.12, $\Psi_{A,B}N_P$ is rank-3. Therefore, $|\mathcal{N}_A \cap \mathcal{N}_B| \geq 3$. By Lemma 7.11, $|\{2i-1, 2i, 2j-1, 2j, 2k-1, 2k\} \cap \{2p-1, 2p, 2q-1, 2q, 2r-1, 2r\}| \geq 3$. Therefore, $|\{i,j,k\} \cap \{p,q,r\}| \geq 2$. ∎

Informally speaking, Lemma 7.12 shows that edges in the rigidity graph correspond to pairs of triangles whose mutual rigidity is evident: they share a side. The lemma can be generalized to higher dimensions: elastic elements are mutually rigid if and only if they share a face. For elasticity, this may be a trivial statement, but it shows that our definition of mutual rigidity indeed captures the domain-specific notion of rigidity.

Figure 7.1 shows a few examples of triangular plane elements and their rigidity graphs. The structures in cases (a) and (b) are not rigid, and the rigidity graph is not connected. Case (c) is rigid, and the rigidity graph is connected. Case (d) is rigid, but the rigidity graph does not show it; the graph is not connected. This shows, again, that connectivity of the rigidity graph is not a necessary condition for rigidity.

8. Rigid sparsifications. Our next goal is to sparsify a matrix A defined as a sum of $\mathbb{S}$-compatible SPSD n-by-n matrices, but without changing $\text{null}(A)$. By sparsify we mean that linear systems of the form $Bz = r$, where B is the sparsification of A, should be easier to solve than linear systems of the form $Ax = b$. In this sense, B is sparser than A if it has a subset of the nonzeros of A, or if the pattern graph of B

has smaller balanced vertex separators than the pattern graph of A [10, 16]. There may be other meanings.

8.1. Dropping vertices from the rigidity graph. Perhaps the most obvious way to sparsify $A = \sum_e A_e$ is to drop some of the element matrices from the assembly. This section gives a condition that guarantees that the subset-sum has the same range and null space as A. The analysis is inductive: it analyzes the effect of dropping one element at a time.

LEMMA 8.1. [illegible] G [illegible] $A_1, A_2, \ldots, A_k$ [illegible] $\mathbb{S}$ [illegible] SPSD n [illegible] n [illegible] $\{H_1, H_2, \ldots, H_\ell\}$ [illegible] G [illegible] $\mathcal{N}_{H_i} \cap \mathcal{N}_{H_j} = \phi$ [illegible] $i \neq j$ [illegible] $A = \sum_{f=1}^{k} A_f$ [illegible]

$$B = \sum_{\substack{f=1 \\ f \neq e}}^{k} A_f = A - A_e$$

[illegible] A_e

1. G [illegible] B [illegible]
2. $\mathcal{N}_A = \mathcal{N}_B$

[illegible] A [illegible] B [illegible]

. By Corollary 3.6 and by the condition $\mathcal{N}_A = \mathcal{N}_B$, it is sufficient to show that A and B are mutually rigid. By Lemma 4.1, $A = B + A_e$ is rigid with respect to B. All that is left to show is that B is rigid with respect to A.

Assume without loss of generality that A_e is in H_1. Let

$$C = \sum_{\substack{A_f \in H_1 \\ f \neq e}} A_f.$$

The first condition of the lemma implies that $H_1 \backslash \{A_e\}$ is a nonempty connected subgraph of G. Therefore, there exists a vertex A_c in $H_1 \backslash \{A_e\}$ such that A_e and A_c are mutually rigid. Applying Theorem 6.3 to $\{A_e, A_c\}$ and to $H_1 \backslash \{A_e\}$ shows that $A_e + A_c$ and C are mutually rigid. Therefore, by Lemma 4.2, C and $C + A_e + A_c$ are mutually rigid. Finally, by Lemma 4.3, C and $C + A_e$ are mutually rigid.

Let $D = \sum_{i=2}^{\ell} \sum_{A_f \in H_i} A_f$. We have that $\mathcal{N}_D \cap \mathcal{N}_C = \phi$ and $\mathcal{N}_D \cap \mathcal{N}_{C+A_e} = \phi$, since $\mathcal{N}_{H_i}$ are pairwise disjoint. By Lemma 4.4, we have that $C + D$ and $C + A_e + D$ are mutually rigid. Therefore, $A = C + A_e + D$ and $B = C + D$ are mutually rigid, so $\text{null}(A) = \text{null}(B)$. □

In particular, if G is connected, then the lemma allows us to drop element matrices only as long as the rigidity graph of the remaining elements remains connected. Clearly, there are cases where we can drop an element matrix that would disconnect the rigidity graph without changing the null space of the sum. In this case dropping the element violates the sufficient condition stated in Lemma 8.1, but without actually changing the null space. For example, dropping $A^{(2,5,6)}$ from the structure shown in Figure 7.1.(c) leads to the structure shown in Figure 7.1.(d), which is also rigid but has a disconnected rigidity graph.

The examples shown in Figure 7.1 (a) and (b) show that the lemma does not hold if the $\mathcal{N}_{H_i}$ are not mutually disjoint. Dropping $A^{(3,4,5)}$ from the structure shown in part (a) of the figure gives the structure shown in (b). The rigidity graphs of both

structures have the same number of connected components, 2, and $\mathcal{N}_A = \mathcal{N}_{A-A^{(3,4,5)}}$. But the null space of the structure in (a) has dimension 4 (rigid body motions and a rotation around p_3), whereas the null space of the structure in (b) has dimension 6 (separate rigid body motions for the two elements).

If we use Lemma 8.1 to construct a preconditioner B by repeatedly dropping element matrices from the sum $A = \sum_i A_i$, the generalized eigenvalues of (B, A) are clearly bounded from above by 1, since for any λ that satisfies $Bx = \lambda Ax$ for an $x \notin \text{null}(A)$ we have

$$\begin{aligned}\lambda &\leq \max_{\substack{x \\ Ax \neq 0}} \frac{x^T B x}{x^T A x} \\ &= \max_{\substack{x \\ Ax \neq 0}} \frac{x^T \left(\sum_{i \in S \subset \{1,\ldots,k\}} A_i \right) x}{x^T \left(\sum_{i=1}^{k} A_i \right) x} \\ &\leq 1.\end{aligned}$$

8.2. Dropping edges from the rigidity graph by fretsaw extensions. We now show and analyze a more sophisticated sparsification technique that drops edges from the rigidity graph. This technique allows us to algebraically express the physical action of cutting slits in a structure without changing its qualitative behavior (its null modes) and its material properties. The technique drops edges from the rigidity graph of a given structure and then modifies the indices of nonzero rows and columns in element matrices to simulate the action of cutting slits along element boundaries where edges have been dropped. Figure 8.1 illustrates the cutting idea and its connection to dropped edges in the rigidity graph. The technique is quite complex to define and analyze.

DEFINITION 8.2. [illegible] extension mapping [illegible] n [illegible] ℓ [illegible] $s = \begin{bmatrix} s_1 & s_2 & \cdots & s_\ell \end{bmatrix}$ [illegible] $1, n$ [illegible] master extension matrix P [illegible] s [illegible] $(n+\ell) \times n$ [illegible] 1 [illegible]

$$P_{ij}^{(s)} = \begin{cases} 1, & i \leq n \text{ and } j = i, \\ 1, & i > n \text{ and } j = s_{i-n}, \\ 0 & \text{otherwise.} \end{cases}$$

[illegible] extension matrix $Q^{(s)}$ [illegible] s [illegible] $(n+\ell) \times n$ [illegible] 1 [illegible] 1 [illegible] $P_{ij}^{(s)} = 0$ [illegible] $Q_{ij}^{(s)} = 0$ [illegible]

In the product $Q^{(s)}A$ of an extension matrix $Q^{(s)}$ and an arbitrary matrix A, each row of the product is either all zeros or a copy of some row of A, and each row of A is mapped to a row of the product. In particular, row i of A is mapped either to row i of the product or to row $n + j$, where $s_j = i$.

The following lemma states some properties of extension matrices and their relation to the projection matrices Ψ_A. We omit its proof.

LEMMA 8.3. [illegible] $P = P^{(s)}$ [illegible] $Q = Q^{(s)}$ [illegible] [illegible] s [illegible] ℓ [illegible] A [illegible] $n \times n$

I_n n n $Q^TQ = P^TQ = Q^TP = I_n$ $\Psi_{QAQ^T} = Q\Psi_A Q^T$

DEFINITION 8.4. $\mathbb{S} \subseteq \mathbb{R}^n$ s $\operatorname{span}\left(P^{(s)}N\right)$ N $\mathbb{S}$ $P^{(s)}\mathbb{S}$

LEMMA 8.5. $P^{(s)}\mathbb{S}$ $P^{(s)}$ $\mathbb{S}$ N N_1 N_2 $\mathbb{S}$ $\operatorname{span}\left(P^{(s)}N_1\right) = \operatorname{span}\left(P^{(s)}N_2\right)$

Proof. Let $x \in \operatorname{span}\left(P^{(s)}N_1\right)$. There exists a vector y such that $x = P^{(s)}N_1 y$. Since $N_1 y \in \mathbb{S}$, there exists a vector z such that $N_1 y = N_2 z$. Therefore, $x = P^{(s)}N_1 y = P^{(s)}N_2 z \in \operatorname{span}(P^{(s)}N_2)$. This implies that $\operatorname{span}\left(P^{(s)}N_1\right) \subseteq \operatorname{span}(P^{(s)}N_2)$. Equality follows by symmetry. ∎

Note that $\operatorname{span}\left(P^{(s)}N\right)$ is essentially the same space as N, but with some coordinates replicated. This observation serves as an intuition for the proofs of the following two lemmas.

The following lemma shows that the extension of an $\mathbb{S}$-compatible SPSD matrix retains its essential properties.

LEMMA 8.6. A $\mathbb{S}$ SPSD $Q = Q^{(s)}$ s QAQ^T SPSD $P^{(s)}\mathbb{S}$

Proof. The matrix QAQ^T is symmetric since A is symmetric. For an arbitrary vector x, let $y = Q^Tx$. We have that $x^TQAQ^Tx = y^TAy \geq 0$, since A is positive semidefinite. This implies that QAQ^T is positive semidefinite.

We now show that QAQ^T is compatible with $P^{(s)}\mathbb{S}$. Let N be a matrix whose columns form a basis for $\mathbb{S}$. It is sufficient to show that $\operatorname{span}(\Psi_{QAQ^T}P^{(s)}N) = \operatorname{enull}(QAQ^T)$.

By Lemma 8.3,

$$\Psi_{QAQ^T}P^{(s)}N = Q\Psi_A Q^T P^{(s)}N = Q\Psi_A N.$$

Therefore, $\operatorname{span}(\Psi_{QAQ^T}P^{(s)}N) = \operatorname{span}(Q\Psi_A N)$. Let E be a matrix whose columns form a basis for $\operatorname{enull}(A)$. Since A is $\mathbb{S}$-compatible, $\operatorname{span}(\Psi_A N) = \operatorname{span}(E)$. Moreover, since Q has full rank, $\operatorname{span}(Q\Psi_A N) = \operatorname{span}(QE)$. Therefore, $\operatorname{span}(\Psi_{QAQ^T}P^{(s)}N) = \operatorname{span}(QE)$.

Now, it is sufficient to show that $\operatorname{span}(QE) = \operatorname{enull}(QAQ^T)$ in order to prove the lemma. In the special case where Q is an $(n+\ell)$-by-n identity matrix, this equality directly follows the fact that $\operatorname{span}(E) = \operatorname{enull}(A)$. In the general case, Q is a row permutation of the $(n+\ell)$-by-n identity matrix, and the equlity holds since renaming the variables does not change anything. ∎

Similarly extended matrices maintain their rigidity relationship.

LEMMA 8.7. $\{A_i\}_{i=1}^k$ $\mathbb{S}$ SPSD $Q_e = Q_e^{(s)}$ $Q_f = Q_f^{(s)}$ s $[Q_e]_{:,j} = [Q_f]_{:,j}$ $j \in \mathcal{N}_{A_e} \cap \mathcal{N}_{A_f}$ $Q_e A_e Q_e^T$ $Q_f A_f Q_f^T$ A_e A_f Q $(n+\ell)$ n $(n+\ell)$ s) (A_e, A_f) $\{A_i\}_{i=1}^k$ (QA_eQ^T, QA_fQ^T) $\{QA_iQ^T\}_{i=1}^k$

. We start by some immediate definitions and observations. Let N be an n-by-ℓ matrix whose columns form a basis for $\mathbb{S}$. Let $P^{(s)}$ be the corresponding master extension matrix. By Lemma 8.6, the matrix $P^{(s)}$ is a basis for $P^{(s)}\mathbb{S}$, and $Q_e A_e Q_e^T$ and $Q_f A_f Q_f^T$ are $P^{(s)}\mathbb{S}$-compatible.

By Lemma 5.12, the matrices A_e and A_f are mutually rigid if and only if $\Psi_{A_e,A_f}N$ is rank-ℓ. The matrix $\Psi_{Q_e A_e Q_e^T, Q_f A_f Q_f^T}P^{(s)}N$ equals the matrix $\Psi_{A_e,A_f}N$. Therefore, $\Psi_{A_e,A_f}N$ is rank-ℓ if and only if $\Psi_{Q_e A_e Q_e^T, Q_f A_f Q_f^T}P^{(s)}N$ is rank-ℓ. Applying Lemma 5.12 again, we get that $\Psi_{Q_e A_e Q_e^T, Q_f A_f Q_f^T}P^{(s)}N$ is rank-ℓ if and only if $Q_e A_e Q_e^T$ and $Q_f A_f Q_f^T$ are mutually rigid. This concludes the proof of the lemma. □

We are particularly interested in certain extensions, described by the following definition.

DEFINITION 8.8. . $\{A_i\}_{i=1}^k$ $\mathbb{S}$ s $\{Q_i\}_{i=1}$ s G $\{A_i\}_{i=1}$ $\hat{G}$ $\{Q_i A_i Q_i^T\}_{i=1}$

- $j \in \mathcal{N}_A$ i $j \in \mathcal{N}_{A_i}$ $[Q_i]_{j,j} = 1$
- $A_{i_1}, \ldots, A_{i_m}$ G $Q_{i_1} A_{i_1} Q_{i_2}^T, \ldots, Q_{i_m} A_{i_m} Q_{i_m}^T$ $\hat{G}$
- $A_{i_1}, \ldots, A_{i_m}$ G $j \in \{i_1, \ldots, i_m\}$ Q_j $(n+\ell)$

$\{Q_i A_i Q_i^T\}_{i=1}^k$ fretsaw extension $\{A_i\}_{i=1}^k$ ( ) Q_i $\mathcal{F}(A)$ $\sum_{i=1}^k Q_i A_i Q_i^T$ fretsaw extension of $A = \sum_{i=1}^k A_i$.

We note that the first fretsaw condition ensures that $\mathcal{N}_A \subseteq \mathcal{N}_{\mathcal{F}(A)}$.

Transforming an extended matrix $B = \sum Q_i^{(s)} A_i \, {Q_i^{(s)}}^T$ back to $A = \sum A_i$ is simple, as shown in the next lemma.

LEMMA 8.9. . $A_1, A_2, \ldots, A_k$ n n P s $\{Q_i\}_{i=1}^k$ s $A = \sum_{i=1}^k A_i$ $B = \sum_{i=1}^k Q_i A_i Q_i^T$ $A = P^T B P$

. By Lemma 8.3, $P^T Q_i = I$, so

$$P^T B P = \sum_{i=1}^k P^T Q_i A_i Q_i^T P = \sum_{i=1}^k A_i = A. \qquad \square$$

DEFINITION 8.10. . B $(n+\ell)$ $(n+\ell)$

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

B_{11} n n B_{22} ℓ ℓ B_{22} $\mathrm{schur}_\ell(B) = B_{11} - B_{12}B_{22}^{-1}B_{21}$ ℓ $\mathrm{schur}(B)$

The following theorem is the main structural result of this section. The theorem characterizes the relationship of the null spaces of a matrix and its fretsaw extension. In particular, it lists conditions that guarantee the preservation of the null space under a fretsaw extension.

Algorithm 1. Pseudocode for the construction of fretsaw spanning-tree extensions.

```
TreeFretsawExtension(𝒜, G)
▷ 𝒜 = {A_1, A_2, ..., A_k} is a collection of 𝕊-compatible SPSD n-by-n matrices
▷ G = (𝒜, E) is the rigidity graph of 𝒜
T ←A spanning tree of G
r ← n
For j ∈ {1, ..., n} do  ▷ in any order
   ▷ Construct column j of Q_1, ..., Q_k
   Set the jth column of Q_1 to be e_j
   V^(j) ← {A_i ∈ 𝒜 | j ∈ 𝒩_{A_i}}  ▷ the elements that are incident on the index j
   E^(j) = {(A_p, A_q) ∈ T | A_p, A_q ∈ V^(j)}
   G^(j) ← (V^(j), E^(j))  ▷ the subgraph of T that is induced by V^(j)
   G_1^(j), G_2^(j), ..., G_{c_j}^(j) ← The connectivity components of G^(j)
    ▷ The order is arbitrary, except that if A_1 ∈ V^(j), then A_1 ∈ G_1^(j)
   For all A_p ∈ G_1^(j), set the jth columns of Q_p to e_j
   For G_i^(j) ∈ {G_2^(j), ..., G_{c_j}^(j)} do  ▷ in any order
      r ← r + 1
      For all A_p ∈ G_i^(j), set the jth columns of Q_p to e_r
   End For
   For i ∈ {1, ..., k} do
      If j ∉ 𝒩_{A_i}, then set the jth columns of Q_i to e_j
   End For
End For
```

THEOREM 8.11. [illegible] $A = \sum_{i=1}^{k} A_i$ [illegible] $A_1, A_2, \ldots, A_k$ [illegible] $\mathbb{S}$ [illegible] n [illegible] n [illegible] Q [illegible] $(n+\ell)$ [illegible] n [illegible] $\mathcal{F}(A)$ [illegible] A [illegible] $\mathcal{N}_A = \{1, \ldots, n\}$ [illegible] $A_1, A_2, \ldots, A_k$ [illegible]:

1. QAQ^T [illegible] $\mathcal{F}(A)$ [illegible]
2. [illegible] $x \in \mathrm{enull}(QAQ^T)$ [illegible] $y \in \mathrm{enull}(\mathcal{F}(A))$ [illegible] $\Psi_{QAQ^T} y = x$
3. $\mathrm{null}(\mathcal{F}(A)) \subseteq \mathrm{null}(QAQ^T)$
4. [illegible] $\mathrm{schur}(\mathcal{F}(A))$ [illegible] $\mathrm{null}(A) = \mathrm{null}(\mathrm{schur}(\mathcal{F}(A)))$

[illegible]. Let $\{Q_i\}_{i=1}^k$ be the collection of the extension matrices used in $\mathcal{F}(A)$. By Lemma 8.6, the matrices in collections $\{Q_i A_i Q_i^T\}_{i=1}^k$ and $\{QA_iQ^T\}_{i=1}^k$ are compatible with $P^{(s)}\mathbb{S}$. By definition, the rigidity graph of $\{Q_iA_iQ_i^T\}_{i=1}^k$ is connected. By Lemma 8.7 and the assumption that the rigidity graph of $\{A_i\}_{i=1}^k$ is connected, the rigidity graph of $\{QA_iQ^T\}_{i=1}^k$ is also connected. By definition, the rigidity graph of $\{QA_iQ^T\}_{i=1}^k$ shares at least one vertex with the rigidity graph of $\{Q_iA_iQ_i^T\}_{i=1}^k$. Therefore, by Lemma 6.3, QAQ^T and $\mathcal{F}(A)$ are mutually rigid.

The second part of the lemma follows the fact that QAQ^T is rigid with respect to $\mathcal{F}(A)$ and that $\mathcal{N}_{QAQ^T} = \mathcal{N}_A \subseteq \mathcal{N}_{\mathcal{F}(A)}$.

The matrix $\mathcal{F}(A)$ is rigid with respect to QAQ^T. We also have that $\mathcal{N}_{QAQ^T} = \mathcal{N}_A \subseteq \mathcal{N}_{\mathcal{F}(A)}$. Therefore, by Lemma 3.5, $\mathrm{null}(\mathcal{F}(A)) \subseteq \mathrm{null}(QAQ^T)$. This proves the third part of the lemma.

The fourth part of the lemma follows from Lemma 3.7 and the fact that QAQ^T and $\mathcal{F}(A)$ are mutually rigid. ∎

8.3. Constructing a fretsaw extension from a spanning tree. We now show a practical way to construct nontrivial fretsaw extensions. The extensions that we build here are essentially as sparse as possible: we can factor $\mathcal{F}(A)$ with no fill. Our simple spanning-tree extensions may not be effective preconditioners (the generalized eigenvalues may be large), but the construction shows that there are efficient algorithms for constructing nontrivial fretsaw extensions.

Let $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$ be a collection of $\mathbb{S}$-compatible symmetric positive semidefinite n-by-n matrices. Let $G = (\mathcal{A}, E)$ be the rigidity graph of $\mathcal{A}$. Without loss of generality, we assume that G is connected (otherwise we repeat the construction for each connected component of G). The construction builds the Q_is by columns. This introduces a slight notational difficulty, since we do not know the number of rows in the Q_is until the construction ends. We use the convention that the columns are tall enough (nk is tall enough) and then chop the Q_is to remove the rows that are zero in all of them. We denote by e_r the rth (long enough) unit vector.

We use a spanning tree T of G to define an extension $\mathcal{F}(A)$. We initialize a variable r to n. This variable stores the index of the last nonzero row in the Q_is. We also initialize the extension mapping $s = \langle\rangle$. The algorithm iterates over the column indices $j \in \{1, \ldots, n\}$ (in any order). In iteration j, we construct column j of $Q_1, \ldots, Q_k$.

We begin iteration j by setting the jth column of Q_1 to e_j. This ensures that Q_1 is an identity matrix, so the third fretsaw condition is automatically satisfied.

We now construct the set $V^{(j)} = \{A_i \in \mathcal{A} | j \in \mathcal{N}_{A_i}\}$ of elements that are incident on the index j. We also construct the subgraph $G^{(j)} = (V^{(j)}, E^{(j)})$ of T that is induced by $V^{(j)}$. We partition $G^{(j)}$ into its connected components and process each component separately. The ordering of the components is arbitrary, except that if $A_1 \in V^{(j)}$, then we process the component containing A_1 first. Let $\{A_{i_1}, A_{i_2}, \ldots, A_{i_m}\} \subseteq V^{(j)}$ be the vertices of the next component to be processed. If this component is the first component of $G^{(j)}$, then we set the jth columns of $Q_{i_1}, \ldots, Q_{i_m}$ to e_j. Otherwise, we increment r, set the jth columns of $Q_{i_1}, \ldots, Q_{i_m}$ to e_r, and concatenate the current index j to the end of s.

This process specifies the jth column of every Q_i such that $j \in \mathcal{N}_{A_i}$. We complete the construction of the Q_is by setting the jth column of every Q_i such that $j \notin \mathcal{N}_{A_i}$ to e_j.

Sometimes the row/column indices of $A = \sum A_i$ have a natural grouping. For example, in problems arising in two-dimensional linear elasticity, each point in the geometry of the discrete structure is associated with two indices, an x-direction index, say, j_1, and a y-direction index, say, j_2. This usually implies that $G^{(j_1)}$ and $G^{(j_2)}$ are identical graphs. In such cases, it may be wise to [illegible] A [illegible]: we order the connected components of $G^{(j_1)}$ and $G^{(j_2)}$ consistently, which means that $[Q_i]_{:,j_1} = e_{j_1}$ if and only if $[Q_i]_{:,j_2} = e_{j_2}$. A consistent extension has a physical interpretation in terms of slits in the material, whereas an inconsistent extension may have no straightforward physical meaning.

Figure 8.1 illustrates the construction of a spanning-tree fretsaw-extended matrix for a structure consisting of linear elastic elements in two dimensions. The figure explains the rationale behind the term fretsaw. A fretsaw is a fine-toothed saw held under tension, designed for cutting thin slits in flat materials, such as sheets of plywood. When applied to two-dimensional elastic structures, like the one shown in Figure 8.1, the spanning-tree fretsaw construction appears to cut the original structure like a fretsaw.

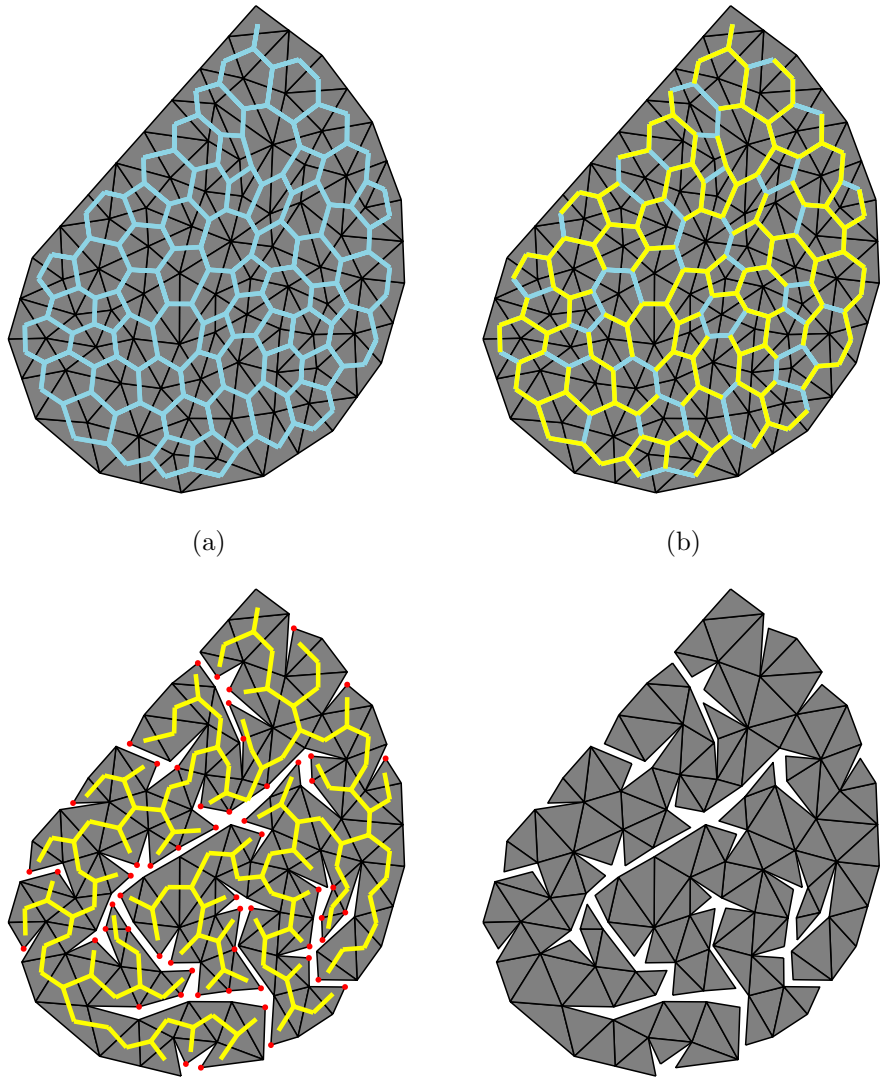


FIG. 8.1. *The construction of a spanning-tree fretsaw-extended matrix.* (a) *The elements of the original structure and its rigidity graph in blue; elements are mutually rigid if and only if they share a side.* (b) *A spanning tree T (yellow) of the rigidity graph.* (c) *The structure induced by the spanning tree; duplicated nodes are marked by red circles. In the illustration, the triangles have been slightly shrunk to show how rigidity relationships have been severed, but the element matrices are only permuted, so they still model the original triangles.* (d) *The fretsaw-extended structure.*

Once the Q_i's are constructed, we form $\mathcal{F}(A) = \sum_{i=1}^{k} Q_i A_i Q_i^T$. The next theorem shows that the Q_i's are extension matrices for some s and that $\mathcal{F}(A)$ is a fretsaw extension of A.

THEOREM 8.12. $\{A_i\}_{i=1}^{k}$ $\mathbb{S}$ SPSD n n s $\{Q_i\}_{i=1}^{k}$ $\mathcal{F}(A) = \sum_{i=1}^{k} Q_i A_i Q_i^T$ A

. We first show that the Q_i's are indeed valid extension matrices for a single s. We construct s step-by-step simultaneously with the construction of the Q_i's. We initialize $s = \langle\rangle$. When we increment the value r, we concatenate the current index j to the end of s. By definition, Q_i contains a nonzero in (r, j) if and only if s contains the value j in its $(r-n)$th position. Therefore, the Q_i's are consistent with $P^{(s)}$. Moreover, by definition, every column j of a matrix Q_i is either e_j or e_r, where $r > n$ and r is unique among the columns of Q_i. Therefore, the Q_i's are valid extension matrices.

Let G be the rigidity graph of $\{A_i\}_{i=1}^k$, and let $\hat{G}$ be the rigidity graph of $\{Q_i A_i Q_i^T\}_{i=1}^k$. We show now that for every connected component $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ in G, the matrices $Q_{i_1}A_{i_1}Q_{i_1}^T, Q_{i_2}A_{i_2}Q_{i_2}^T, \ldots, Q_{i_m}A_{i_m}Q_{i_m}^T$ form a connected component in $\hat{G}$ and vice versa. Let $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ be a connected component of G. Let T be the spanning tree used to create $Q_{i_1}, Q_{i_2}, \ldots, Q_{i_m}$. Let A_p and A_q be two matrices adjacent in T. For every index $j \in \mathcal{N}_{A_p} \cap \mathcal{N}_{A_q}$, A_p and A_q belong to the same connected component of $G^{(j)}$. Therefore, Q_p and Q_j coincide on their jth column. Therefore, by Lemma 8.7, $Q_p A_p Q_p^T$ and $Q_q A_q Q_q^T$ are mutually rigid. Therefore, $Q_{i_1}A_{i_1}Q_{i_1}^T, Q_{i_2}A_{i_2}Q_{i_2}^T, \ldots, Q_{i_m}A_{i_m}Q_{i_m}^T$ is a connected component in $\hat{G}$. In a similar manner, if $Q_{i_1}A_{i_1}Q_{i_1}^T, Q_{i_2}A_{i_2}Q_{i_2}^T, \ldots, Q_{i_m}A_{i_m}Q_{i_m}^T$ form a connected component in $\hat{G}$, by Lemma 8.7, $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ is a connected component in G.

There are two additional properties that need to be verified in order to show that $\mathcal{F}(A)$ is a fretsaw extension. By definition, the construction ensures that there is at least one Q_{i_1} for every connected component which is an $(n+l)$-by-n identity matrix. The second property that we need to show is that $\mathcal{N}_A \subseteq \mathcal{N}_{\mathcal{F}(A)}$. Let $j \in \mathcal{N}_A$. Let A_p be a matrix in the connected component of $G^{(j)}$ that was processed first. By definition, column j of Q_p contains e_j. Therefore, $j \in \mathcal{N}_{Q_p A_p Q_p^T}$. Lemma 2.6 ensures that $j \in \mathcal{N}_{\mathcal{F}(A)}$. □

8.4. Perfect elimination orderings for spanning-tree fretsaw extensions. The spanning-tree fretsaw construction is motivated by an elimination ordering that guarantees that all the fill occurs within the nonzero structure of the element matrices. If $[A_e]_{i,j} \neq 0$ for all $i, j \in \mathcal{N}_A$, this ordering is a no-fill ordering of $\mathcal{F}(A)$.

The analysis in the proof is closely related to the analysis of clique trees. The spanning-tree fretsaw is a tree of cliques that satisfies the clique-intersection property defined by Blair and Peyton [3]. If the elements were guaranteed to be , cliques, then Theorem 3.2 of [3] would guarantee a no-fill elimination ordering. But we are not certain whether element matrices are always maximal cliques in the graph of the assembled matrix, so [3, Theorem 3.2] is not directly applicable.

LEMMA 8.13. . $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$ $\mathbb{S}$ SPSD n n $\mathcal{F}(A)$ $A = \sum_{i=1}^k A_i$. Γ $L_{i,j}$ L $\Gamma\mathcal{F}(A)\Gamma^T$ $i, j \in \mathcal{N}_{\Gamma Q_e A_e Q_e^T \Gamma^T}$ e (e i j)

. We root the spanning tree T of the rigidity graph at A_1 and take ϕ to be a postorder of this rooted tree. That is, ϕ is an ordering of the element matrices in which the leaves of the rooted tree appear first, followed by parents of leaves, etc. We construct an elimination ordering γ incrementally. Initially, $\gamma = \langle\rangle$ is an empty ordering. Let A_e be the next unprocessed element matrix in ϕ, and let A_f be the parent of A_e in the rooted tree (if it has a parent). Let $\{i_1, \ldots, i_m\}$ be the indices in

$$\mathcal{N}_{Q_e A_e Q_e^T} \setminus \left(\mathcal{N}_{Q_f A_f Q_f^T} \bigcup \gamma\right)$$

(with γ taken to be a set of already ordered indices). We concatenate $\langle i_1, \ldots, i_m \rangle$ to γ in an arbitrary order. The permutation matrix Γ is the matrix that corresponds to γ. That is, $\Gamma_{i,\gamma_i} = 1$.

Now that we have specified Γ, we show that it limits fill as claimed.

A. Let $i \in \{1, \ldots, n, n+1, \ldots, n+\ell\}$. Let j be the column in $P^{(s)}$ such that $P^{(s)}_{i,j} \neq 0$ (recall that every row in $P^{(s)}$ contains exactly one nonzero). We denote by $G^{(j)}_i$ the connected component of $G^{(j)}$ in which j is mapped to i. The graph $G^{(j)}_i$ is a connected subgraph of $G^{(j)}$ which is an induced subgraph of a rooted tree. Therefore, $G^{(j)}_i$ is itself a rooted tree. We claim that i is added to γ during the processing of the root A_h of $G^{(j)}_i$.

A. We first show that if i is added to γ, then it is added during the processing of the root of $G^{(j)}_i$. Let A_e be the element during the processing of which we add i to γ. We first show that $A_e \in G^{(j)}_i$. Clearly, $i \in \mathcal{N}_{Q_e A_e Q_e^T}$. Therefore, $j \in \mathcal{N}_{A_e}$. By the definition of $G^{(j)}$, $A_e \in G^{(j)}$, and by the definition of $G^{(j)}_i$, $A_e \in G^{(j)}_i$. Now suppose for contradiction that A_e is not the root of $G^{(j)}_i$. Then A_e has a parent A_f in $G^{(j)}_i$. Because A_f is in $G^{(j)}_i$, $i \in \mathcal{N}_{Q_f A_f Q_f^T}$, so the algorithm would not have added i to γ during the processing of A_e. Therefore, A_e the root of $G^{(j)}_i$.

To complete the proof of Claim A, we show that i is added to γ. Suppose for contradiction that it is not. When we process the root A_h of $G^{(j)}_i$, $i \notin \gamma$. But i cannot be in $\mathcal{N}_{Q_f A_f Q_f^T}$, where A_f is the parent of A_h in the global rooted tree. If it was, then $j \in \mathcal{N}_{A_f}$, so A_f would be in $G^{(j)}$, and because it is connected to A_h, it must also be in $G^{(j)}_i$. But A_h is the root of $G^{(j)}_i$, so $i \notin \mathcal{N}_{Q_f A_f Q_f^T}$. Therefore, i is added to γ. This concludes the proof of Claim A.

B. Just before A_f is processed, γ is exactly the set

$$\gamma = \left\{ i : i \in \mathcal{N}_{Q_e A_e Q_e^T} \text{ for some } A_e \text{ that appears before } A_f \text{ in } \phi \right\}$$
$$\setminus \left(\mathcal{N}_{Q_f A_f Q_f^T} \cup \left\{ i : i \in \mathcal{N}_{Q_g A_g Q_g^T} \text{ for some } A_g \text{ that appears after } A_f \text{ in } \phi \right\} \right).$$

B. The claim follows by induction from the process of constructing γ and from the fact that ϕ is a postorder of the rooted tree.

C. If $L_{\hat r,\hat c} \neq 0$ then $\hat r, \hat c \in \mathcal{N}_{\Gamma Q_e A_e Q_e^T \Gamma^T}$ for some e.

C. If $L_{\hat r,\hat c} \neq 0$, then either $\left[\Gamma \mathcal{F}(A) \Gamma^T\right]_{\hat r,\hat c} \neq 0$ or there is some $\hat\imath < \hat r, \hat c$ such that $L_{\hat r,\hat\imath} \neq 0$ and $L_{\hat c,\hat\imath} \neq 0$. The first condition cannot violate Claim B, because if $\left[\Gamma \mathcal{F}(A) \Gamma^T\right]_{\hat r,\hat c} \neq 0$ then there is some e such that $\left[\Gamma Q_e A_e Q_e^T \Gamma^T\right]_{\hat r,\hat c} \neq 0$, so $\hat r, \hat c \in \mathcal{N}_{\Gamma Q_e A_e Q_e^T \Gamma^T}$.

If for some $\hat r$ and $\hat c$ we have $L_{\hat r,\hat c} \neq 0$ because of the second condition, then let $\hat\imath$ be the minimal index such that $L_{\hat r,\hat\imath} \neq 0$ and $L_{\hat r,\hat\imath} \neq 0$ $(\hat r, \hat c)$. This definition of $\hat\imath$ guarantees that $\left[\Gamma \mathcal{F}(A) \Gamma^T\right]_{\hat r,\hat\imath} \neq 0$ and $\left[\Gamma \mathcal{F}(A) \Gamma^T\right]_{\hat c,\hat\imath} \neq 0$. Define $i = \gamma_{\hat\imath}$, $r = \gamma_{\hat r}$, and $c = \gamma_{\hat c}$.

Let A_f be the element during the processing of which i was added to γ. Because $\hat\imath < \hat r, \hat c$, when i is added to γ, r and c are not yet in γ. We claim that r and c are in $\mathcal{N}_{Q_f A_f Q_f^T}$; if this is true, Claim C holds. Suppose for contradiction that $r \notin \mathcal{N}_{Q_f A_f Q_f^T}$. By Claim B, $r \in \mathcal{N}_{A_g}$ for some A_g that appears after A_f in ϕ. Because $\left[\Gamma \mathcal{F}(A) \Gamma^T\right]_{\hat r,\hat\imath} \neq 0$ and because A_g is symmetric, we must also have $i \in \mathcal{N}_{A_g}$.

But this implies that i cannot be added to γ when A_f is processed (by Claim B). This concludes the proof of Claim C and the entire proof. ∎

8.5. Quantitative analysis. Lemma 8.11 showed that if the rigidity graph of a finite-element matrix $A = \sum A_e$ is connected and if $\mathrm{schur}(\mathcal{F}(A))$ exists for a fretsaw extension $\mathcal{F}(A)$, then A and $\mathrm{schur}(\mathcal{F}(A))$ have the same range and null space. We now strengthen this result and show that the generalized eigenvalues of $(\mathrm{schur}(\mathcal{F}(A)), A)$ are bounded from above by 1. We note that $\mathrm{schur}(\mathcal{F}(A))$ can be implicitly used as a preconditioner; in the preconditioning step of an iterative linear solver, we can solve a linear system whose coefficient matrix is $\mathcal{F}(A)$—not $\mathrm{schur}(\mathcal{F}(A))$ [11, section 3.3]. In particular, the previous section showed that we can factor a spanning-tree fretsaw extension $\mathcal{F}(A)$ with essentially no fill.

LEMMA 8.14. Let $A_1, A_2, \ldots, A_k$ be matrices in $\mathbb{S}^+$, each SPSD n-by-n, and let $A = \sum_{i=1}^{k} A_i$. Let $\mathcal{F}(A)$ be a fretsaw extension of A such that $\mathrm{schur}(\mathcal{F}(A))$ exists. Let λ be a finite generalized eigenvalue of $(\mathrm{schur}(\mathcal{F}(A)), A)$. Then $\lambda \leq 1$.

Proof. We partition $\mathcal{F}(A)$ into

$$\mathcal{F}(A) = \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix},$$

where B_{11} is n-by-n and B_{22} is ℓ-by-ℓ. By the assumption that $\mathrm{schur}(\mathcal{F}(A))$ exists, B_{22} is symmetric positive definite. In this notation, $\mathrm{schur}(\mathcal{F}(A)) = B_{11} - B_{12}B_{22}^{-1}B_{12}^T$. Let P be the $(n+\ell)$-by-n master extension matrix corresponding to the fretsaw extension $\mathcal{F}(A)$ and let Q be the $(n+\ell)$-by-n identity matrix.

Let $\lambda_{\max}$ be the maximal finite generalized eigenvalue of the pencil $(\mathrm{schur}(\mathcal{F}(A)), A)$ and let x be the corresponding eigenvector. We let

$$\hat{x} = \begin{bmatrix} x \\ -B_{22}^{-1}B_{21}x \end{bmatrix}$$

and multiply it by $\mathcal{F}(A)$:

$$\begin{aligned} \mathcal{F}(A)\hat{x} &= \begin{bmatrix} B_{11}x + B_{12}(-B_{22}^{-1}B_{21}x) \\ B_{21}x + B_{22}(-B_{22}^{-1}B_{21}x) \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{schur}(\mathcal{F}(A))x \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \lambda_{\max}Ax \\ 0 \end{bmatrix} \\ &= \lambda_{\max}QAQ^T\hat{x}\,. \end{aligned}$$

Multiplying both sides by $\hat{x}^T$, we obtain $\hat{x}^T\mathcal{F}(A)\hat{x} = \lambda_{\max}\hat{x}^TQAQ^T\hat{x}$.

We now show that $\hat{x}^T\mathcal{F}(A)\hat{x} \leq \hat{x}^TQAQ^T\hat{x}$. For a length-$\ell$ vector y, define the function

$$f(y) = \begin{bmatrix} x^T & y^T \end{bmatrix} \mathcal{F}(A) \begin{bmatrix} x \\ y \end{bmatrix}.$$

We note that $f(-B_{22}^{-1}B_{21}x) = \hat{x}^T\mathcal{F}(A)\hat{x}$. For an arbitrary y,

$$\begin{aligned}
\begin{bmatrix} x^T & y^T \end{bmatrix} \mathcal{F}(A) \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} x^T & y^T \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\
&= x^T B_{11} x + y^T B_{12}^T x + x^T B_{12} y + y^T B_{22} y \\
&= x^T \left(B_{11} - B_{12} B_{22}^{-1} B_{12}^T \right) x + x^T B_{12} B_{22}^{-1} B_{12}^T x \\
&\quad + y^T B_{12}^T x + x^T B_{12} y + y^T B_{22} y \\
&= x^T \left(B_{11} - B_{12} B_{22}^{-1} B_{12}^T \right) x \\
&\quad + \left(y + B_{22}^{-1} B_{12}^T x \right)^T B_{22} \left(y + B_{22}^{-1} B_{12}^T x \right).
\end{aligned}$$

Because B_{22} is positive definite, $f(y)$ is minimized at $y = -B_{22}^{-1}B_{12}^T x$.

By Lemma 8.9, $\hat{x}^T Q A Q^T \hat{x} = \hat{x}^T Q (P^T \mathcal{F}(A) P) Q^T \hat{x} = x^T P^T \mathcal{F}(A) P x$. By the definition of a master extension matrix, the vector Px contains the vector x in its first n coordinates, so $Px = \begin{bmatrix} x^T & z^T \end{bmatrix}$ for some z and $x^T P^T \mathcal{F}(A) P x = f(z)$. Since $-B_{22}^{-1}B_{12}^T x$ minimizes f,

$$\hat{x}^T Q A Q^T \hat{x} = x^T P^T \mathcal{F}(A) P x = f(z) \geq f(-B_{22}^{-1} B_{12}^T x) = \hat{x}^T \mathcal{F}(A) \hat{x}.$$

This implies that $\lambda_{\max} \leq 1$ and concludes the proof of the lemma. □

9. Numerical examples. In this section we present experimental results that indicate that fretsaw-tree sparsifications can be used as preconditioners. We do not claim that they are particularly effective. Our only goal in this section is to demonstrate that fretsaw-tree sparsifications can be used computationally as preconditioners. The results presented in this section also suggest that the qualitative convergence behavior of fretsaw-extension preconditioners is similar to that of Vaidya's preconditioners when applied to weighted Laplacians [8].

Figure 9.1 shows convergence results for an iterative solver (preconditioned conjugate gradients) with a fretsaw-tree preconditioner. The figure shows results for two different physical two-dimensional problems that we discretized on the same triangulated mesh. One problem was a Poisson problem and the other a linear elasticity problem, both with constant coefficients and with Neumann (natural) boundary conditions. In each case, we constrained one or three unknowns belonging to a single triangle to transform the coefficient matrix into a nonsingular one.

Each graph shows convergence results for three conjugate-gradient solvers: with no preconditioning, with no-fill incomplete Cholesky preconditioning (denoted [illegible] (0) in the graphs), and with fretsaw-tree preconditioning. The fretsaw trees for the two problems are different, of course, because the rigidity graphs are different. We chose to compare the fretsaw-tree preconditioner with a no-fill incomplete Cholesky preconditioner because both are equally sparse.

Figure 9.2 shows similar plots for two three-dimensional problems defined on a cylinder, a Poisson problem, and a linear elastic problem. The material is isotropic, but the coefficients are variable; the material constant is $1 + 10x + 1000y + 1000z$, where $x \in [0, 35]$ and $y, z \in [0, 0.2]$. The tree that we constructed was a maximum spanning tree, where the weights were chosen heuristically to reflect the norm and condition of element matrices.

The results show that fretsaw trees can be used as preconditioners. The experiments are too limited to fully judge them, but the experiments do indicate that they are not worse than another no-fill preconditioner. Two other observations on the graphs are (1) the fretsaw is better than incomplete Cholesky on the two-dimensional Poisson problem, but the two preconditioners are comparable on the other problems, and (2) the steady linear convergence behavior of the fretsaw trees is similar to the convergence behavior of Vaidya's preconditioners on weighted Laplacians [8].

10. Concluding remarks. To keep the paper readable and of reasonable length, we have omitted from it several topics, which we plan to cover in other papers.

- Element matrices that represent boundary conditions. In much of this paper, we have assumed that all the element matrices are compatible with $\mathbb{S}$. This means, in particular, that the element matrix is singular. In many practical computations, boundary conditions are added to remove the singularity. We kept the discussion focused on singular matrices to reduce clutter. We plan to explore the handling of boundary conditions in a future paper.
- Fretsaw constructions other than spanning-tree fretsaw extensions. Previous work on combinatorial preconditioners indicates, both theoretically and experimentally, that tree and tree-like preconditioners are not effective; aug-

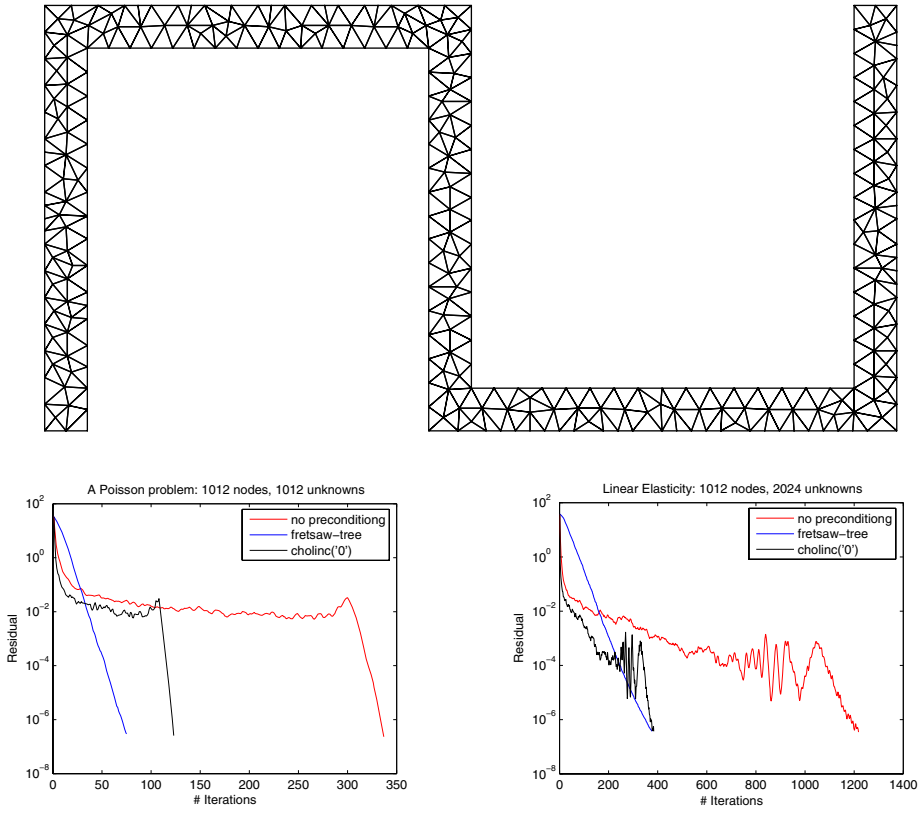


FIG. 9.1. *A triangularization of a two-dimensional domain (top) and convergence plots for two problems discretized on this domain. The triangularization used in the plots is finer than the one shown in the top part of the figure. The graph on the left shows the convergence of iterative solvers on a discretization of a Poisson problem, and the graph on the right shows convergence on a linear elasticity problem, both with constant coefficients.*

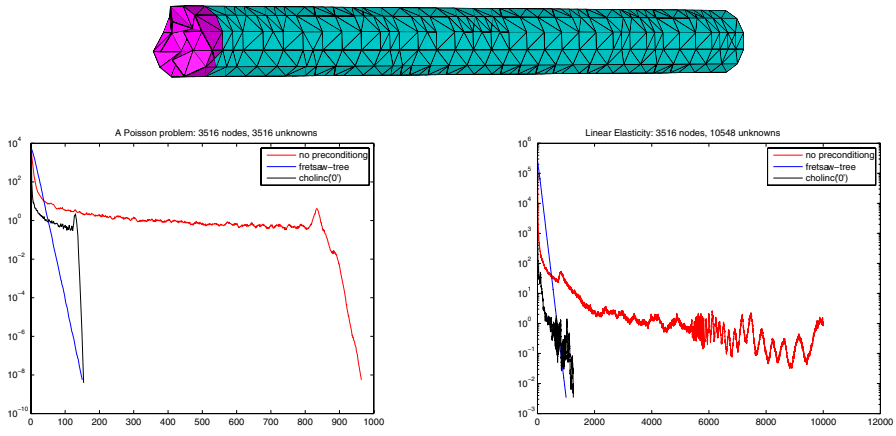


FIG. 9.2. *A tetrahedral meshing of a three-dimensional cylinder (top) and convergence plots for two problems discretized on a similar but longer cylinder (bottom). The leftmost part of the cylinder was removed to reveal the irregular meshing inside.*

mented trees and other constructions usually work better. We have developed augmented spanning-tree fretsaw extension algorithms for Laplacians, but this construction is beyond the scope of this paper.

In addition, there are several interesting problems that we have not yet solved. The most interesting one is proving lower bounds on the generalized eigenvalues of $(\text{schur}(\mathcal{F}(A)), A)$ and finding fretsaw constructions that ensure that this bound is not too small. A particularly interesting question is whether this can be done by assigning weights to the edges of the rigidity graph.

Another question is what other results from spectral graph theory can be generalized to finite-element matrices as defined in this paper, and whether the rigidity graph, perhaps weighted, would be useful in such generalizations.

Acknowledgments. Thanks to the two anonymous referees for numerous comments and suggestions.

REFERENCES

[1] N. ALON AND V. D. MILMAN, λ_1, *isoperimetric inequalities for graphs, and superconcentrators*, J. Combin. Theory Ser. B, 38 (1985), pp. 73–88.

[2] M. BERN, J. R. GILBERT, B. HENDRICKSON, N. NGUYEN, AND S. TOLEDO, *Support-graph preconditioners*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 930–951.

[3] J. R. S. BLAIR AND B. PEYTON, *An introduction to chordal graphs and clique trees*, in Graph Theory and Sparse Matrix Computation, A. George, J. R. Gilbert, and J. W. H. Liu, eds., Springer-Verlag, New York, 1993, pp. 1–29.

[4] E. G. BOMAN, D. CHEN, B. HENDRICKSON, AND S. TOLEDO, *Maximum-weight-basis preconditioners*, Numer. Linear Algebra Appl., 11 (2004), pp. 695–721.

[5] E. G. BOMAN AND B. HENDRICKSON, *Support theory for preconditioning*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 694–717.

[6] E. G. BOMAN, B. HENDRICKSON, AND S. A. VAVASIS, *Solving Elliptic Finite Element Systems in Near-Linear Time with Support Preconditioners*, http://arxiv.org/abs/cs/0407022 (2004).

[7] J. CHEEGER, *A lower bound for the smallest eigenvalue of the Laplacian*, in Problems in Analysis, R. C. Gunning, ed., Princeton University Press, Princeton, NJ, 1970, pp. 195–199.

[8] D. CHEN AND S. TOLEDO, *Vaidya's preconditioners: Implementation and experimental study*, Electron. Trans. Numer. Anal., 16 (2003), pp. 30–49.

[9] M. FIEDLER, *Algebraic connectivity of graphs*, Czechoslovak Math. J., 23 (1973), pp. 298–305.

[10] J. R. Gilbert and R. E. Tarjan, *The analysis of a nested dissection algorithm*, Numer. Math., 50 (1987), pp. 377–404.
[11] K. D. Gremban, *Combinatorial Preconditioners for Sparse, Symmetric, Diagonally Dominant Linear Systems*, Ph.D. thesis, Technical report CMU-CS-96-123, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1996.
[12] B. Hendrickson, *Conditions for unique graph realizations*, SIAM J. Comput., 21 (1992), pp. 65–84.
[13] H. Imai, *On combinatorial structures of line drawings of polyhedra*, Discrete Appl. Math., 10 (1985), pp. 79–92.
[14] D. J. Jacobs and B. Hendrickson, *An algorithm for two-dimensional rigidity percolation: The pebble game*, J. Comput. Phys., 137 (1997), pp. 346–365.
[15] G. Laman, *On graphs and rigidity of plane skeletal structures*, J. Engrg. Math., 4 (1970), pp. 331–340.
[16] R. J. Lipton, D. J. Rose, and R. E. Tarjan, *Generalized nested dissection*, SIAM J. Numer. Anal., 16 (1979), pp. 346–358.
[17] L. Lovász and Y. Yemini, *On generic rigidity in the plane*, SIAM J. Algebraic Discrete Methods, 3 (1982), pp. 91–98.
[18] B. M. Maggs, G. L. Miller, O. Parekh, R. Ravi, and S. L. M. Woo, *Solving Symmetric Diagonally Dominant Systems by Preconditioning*, unpublished manuscript. Available online at http://www.cs.cmu.edu/∼bmm (2002).
[19] D. Spielman and S.-H. Teng, *Solving sparse, symmetric, diagonally dominant linear systems in time* $O(m^{1.31})$, in Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, Cambridge, MA, 2003, pp. 416–427.
[20] L. G. Valiant, *Graph theoretic arguments in low-level complexity*, in Proceedings of the 6th Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Comput. Sci. 53, Springer-Verlag, New York, 1977, pp. 162–176.
[21] W. Whiteley, *Some matroids from discrete applied geometry*, in Matroid Theory, Contemp. Math. 197, J. O. J. Bonin and B. Servatius, eds., AMS, Providence, RI, 1996, pp. 171–311.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 41–55

GENERALIZED EIGENVALUES OF NONSQUARE PENCILS WITH STRUCTURE*

PABLO LECUMBERRI†, MARISOL GÓMEZ†, AND ALFONSO CARLOSENA‡

Abstract. This work deals with the generalized eigenvalue problem for nonsquare matrix pencils $\mathbf{A} - \lambda\mathbf{B}$ such that matrices $\mathbf{A}, \mathbf{B} \in \mathcal{M}_{\mathbb{C}}(m \times n)$ show a given structure. More precisely, we assume they result from removing the first row of some matrix $\mathbf{G} \in \mathcal{M}_{\mathbb{C}}((m+1), n)$ in the case of $\mathbf{A}$, and its last row in the case of $\mathbf{B}$. This structured generalized eigenvalue problem can be found in signal processing methods and in the numerical computation of the greatest common divisor (GCD) of polynomials. Traditional methods for solving the problem $(\mathbf{A} - \lambda\mathbf{B})\,\mathbf{v} = \mathbf{0}$ do not yield valid solutions when the data are not exact, as is often the case in real applications. In this work we adopt a minimal perturbation approach. Taking into account the structure of the matrices involved, we develop a simple algorithm for the computation of the generalized eigenvalues.

Key words. nonsquare pencils, generalized eigenvalue, pseudospectra

AMS subject classifications. 15A18, 15A22, 65F10, 65F15

DOI. 10.1137/060669267

1. Introduction.

1.1. Nonsquare matrix pencils with structure. Given two matrices $\mathbf{A}$, $\mathbf{B}$ of the same dimensions, the set $\{\mathbf{A} - \lambda\mathbf{B};\ \lambda \in \mathbb{C}\}$ constitutes a pencil. The generalized eigenvalues of the matrix pencil are those values of λ for which there exist vectors $\mathbf{v}$ different from zero, called generalized eigenvectors, such that the pair $(\lambda, \mathbf{v})$ satisfies

$$(\mathbf{A} - \lambda\mathbf{B}) \cdot \mathbf{v} = \mathbf{0}.$$

The computation of the generalized eigenvalues has been regarded as a tool of great importance in engineering problems for decades. See [6] for applications in linear systems theory.

Matrices $\mathbf{A}$ and $\mathbf{B}$ form a regular matrix pencil $(\mathbf{A} - \lambda\mathbf{B})$ if they are square and the characteristic polynomial $p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{B})$ is only zero for a finite number of values of λ. The generalized eigenvalue problem for this case has been extensively studied. In [12] there can be found some methods to solve it. The problem is more difficult when the matrices are rectangular or $(\mathbf{A} - \lambda\mathbf{B})$ is square and singular for all values of λ. A matrix pencil built with this kind of matrices is known as a singular matrix pencil and its set of eigenvalues can be finite, empty, or infinite.

It is common to find applications, especially in the field of signal processing, that involve rectangular matrices $\mathbf{A}, \mathbf{B} \in \mathcal{M}_{\mathbb{C}}(m, n)$, $m > n$. Additional measurements, which should allow more accurate estimations, add more rows to matrices $\mathbf{A}$ and $\mathbf{B}$. Besides, matrices $\mathbf{A}$ and $\mathbf{B}$ often show some structure that must be taken into account for a correct treatment of the problem, or that may ease the complexity of

*Received by the editors September 7, 2006; accepted for publication (in revised form) by B. T. Kågström May 10, 2007; published electronically January 25, 2008. This work was supported under grants DPI-2003-08637-C01 and MTM-2004-08219-C02-01 by Dirección General de Investigación, Ministerio de Educación y Ciencia of Spain.

http://www.siam.org/journals/simax/30-1/66926.html

†Mathematics Department, Universidad Pública de Navarra, 31006 Pamplona, Navarra, Spain (pablo.lecumberri@unavarra.es, marisol@unavarra.es).

‡Electrical and Electronic Engineering Department, Universidad Pública de Navarra, 31006 Pamplona, Navarra, Spain (carlosen@unavarra.es).

the computation of the generalized eigenvalues. In this work we consider the following structure: We assume that matrix $\mathbf{A}$ is the result of removing the first row of some matrix $\mathbf{G} \in \mathcal{M}_{\mathbb{C}}(m+1, n)$, while matrix $\mathbf{B}$ results from removing the last row of $\mathbf{G}$. These structured nonsquare matrix pencils appear, for example, in some important signal processing applications [13, 9, 17] (in some of these examples, matrices $\mathbf{A}$, $\mathbf{B}$ also show some other structure, namely the Hankel structure of matrix $\mathbf{G}$; this is not considered in this paper), and in the computation of the greatest common divisor of polynomials [14, 15], which in turn has application in diverse fields, such as network theory, automatic control, and computer-aided geometric design.

1.2. The use of the Kronecker canonical form (KCF). The Jordan canonical form of a square matrix, which describes its eigenvalues and invariant subspaces, can be extended to matrix pencils. A pencil $(\mathbf{A} - \lambda\mathbf{B})$ is similar to its KCF [11], which is a pseudodiagonal matrix whose diagonal building blocks are Jordan blocks, with finite and infinite eigenvalues, and, in the case of singular pencils, singular blocks. The development of algorithms for the computation of the KCF has been a matter of research for decades. Nowadays, numerically stable routines are available (see [1] and the references therein). However, the computation of the KCF is an ill-posed problem, in the sense that small perturbations to the pencil may yield a different KCF, probably a generic one, which only contains singular blocks. This feature hampers the application of the KCF to singular pencils derived from imperfectly known data, which is common in fields such as signal processing.

Previous study of perturbations in singular matrix pencils has been carried out under the light of the KCF. Stewart [19] bounds the perturbation on generalized eigenvalues and eigenvectors implicitly assuming that the perturbation on the pencil does not change the structure of the KCF. A similar assumption is used in [5] to bound the change in deflating subspaces of perturbed singular matrix pencils. In that paper, Demmel and Kågström also consider the regular case. Boley in [2] gives bounds for the perturbation that must be applied to $\mathbf{A}$ for the pencil $(\mathbf{A} - \lambda\mathbf{B})$ to have a regular part in its KCF. Edelman, Elmroth, and Kågström in [8] give bounds on the perturbation that makes a pencil less generic. These works give bounds to perturbations, which may be regarded as important information for some linear systems applications, but do not address the computation of the eigenvalues of the perturbed pencils, which in turn would be useful for many other real-world applications.

1.3. Contribution and structure of the paper. The goal of this paper is to propose a numerical method for finding generalized eigenvalues of nonsquare, and thus singular, matrix pencils. Aiming at its application to problems with real data, the method must be robust with respect to perturbations on the matrices of the pencil. This requirement leads us to discard matrix decompositions or transformations that reveal the structure of the pencil. We adopt the minimal perturbation approach (MPA) for the development of the numerical method. This paper can be seen as a follow-up to [3] and [4] which considers a particular structure for the matrix pencil. In those papers, the MPA for the eigenproblem of singular matrix pencils is presented. Roughly speaking, it consists of minimizing the norm of the perturbation that must be applied to a pencil so that it has a fixed number of generalized eigenvalues. The main contribution of this paper, apart from adapting the results in [4] to the considered structure, is the transformation of the constrained optimization problem of the MPA into an unconstrained one with a low-dimensional parameter vector when only one eigenvalue is assumed to exist. Well-known optimization techniques [16] can be applied to solve it, outperforming the numerical algorithm proposed in [3] in terms of

convergence rate and speed.

The paper is organized as follows: Section 2 reviews the MPA presented in [3] and [4] and its relation to pseudospectra. In section 3 the structure considered in this paper for the matrices that form the matrix pencil is introduced and incorporated to the constrained optimization problem that lies at the heart of the MPA. Then, it is shown to be equivalent to a simpler optimization problem and, in the case of seeking a single eigenvalue, it is further transformed into an unconstrained optimization problem that allows the use of simple numerical algorithms. Section 4 includes an example to show the differences between considering the structure of the matrix pencil and not doing so. Finally, we conclude the paper highlighting the main contribution and future research lines.

1.4. Notation. Matrices are denoted by boldface upper-case letters, such as $\mathbf{A}$, whereas boldface lower-case letters, such as $\mathbf{v}$, stand for column vectors. Scalars and polynomials are in roman typography, as λ and $p(x)$, respectively. $\mathbf{A}^{\mathrm{T}}$ stands for the transpose of matrix $\mathbf{A}$, $\mathbf{A}^{\mathrm{H}}$ stands for the conjugate transpose, and λ^* denotes the complex conjugate of scalar λ. The 2-norm and the Frobenius norm of matrices and vectors are denoted by $\|\cdot\|_2$ and $\|\cdot\|_F$, respectively.

The inner product $\langle \mathbf{u}, \mathbf{v} \rangle$ is defined as $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^{\mathrm{H}} \cdot \mathbf{v}$. Either notation will be used throughout the text. The projection of vector $\mathbf{u}$ onto the range of $\mathbf{A}$ is denoted by $P_{\mathbf{A}}(\mathbf{u}) = \mathbf{P}_{\mathbf{A}} \cdot \mathbf{u}$, where $\mathbf{P}_{\mathbf{A}}$ is the corresponding matrix projection operator. Matrix $\mathbf{P}_{\mathbf{A}}^{\perp}$ is the projection operator over the complement subspace of the range of $\mathbf{A}$. $\mathbf{0}$ stands for a row or column vector whose elements are all equal to zero. Its actual dimensions will become clear from the context.

The ith singular value of a matrix $\mathbf{M} \in \mathcal{M}_{\mathbb{C}}(m,n)$ is denoted by $\sigma_i(\mathbf{M})$, with $1 \leq i \leq \min(m.n)$. The singular values are assumed to be in decreasing order, i.e., $\sigma_1(\mathbf{A}) \leq \sigma_2(\mathbf{A}) \leq \cdots \leq \sigma_{\min(m,n)}(\mathbf{A})$.

2. Pseudospectra and the minimal perturbation approach. In some problems, particularly in singular matrix pencils, the eigenvalues do not change continuously with perturbations of the matrices involved. In these situations, the pseudospectra has proved to be a more useful tool to gain the insight usually provided by eigenvalues.

The spectra of a square matrix $\mathbf{A} \in \mathcal{M}_{\mathbb{C}}(m,m)$ is defined as its set of eigenvalues:

$$\Lambda(\mathbf{A}) = \{ z \in \mathbb{C} : \exists \mathbf{u} \neq \mathbf{0}, \mathbf{A} \cdot \mathbf{u} = z \cdot \mathbf{u} \}.$$

The ϵ-pseudospectra of $\mathbf{A}$, $\Lambda_\epsilon(\mathbf{A})$, includes those elements of $\mathbb{C}$ that are eigenvalues of a matrix obtained from $\mathbf{A}$ by a perturbation of Frobenius norm at most ϵ [20]:

$$\Lambda_\epsilon(\mathbf{A}) = \{ z \in \mathbb{C} : z \in \Lambda(\mathbf{A}+\mathbf{E}), \|\mathbf{E}\|_F \leq \epsilon \}.$$

The spectra of $\mathbf{A}$ is obtained as the ϵ-pseudospectra when $\epsilon = 0$.

The extension to regular matrix pencils is treated in several papers; [7, 10] are some of them. Given two matrices $\mathbf{A}, \mathbf{B} \in \mathcal{M}_{\mathbb{C}}(m,m)$, the pseudospectra of the pair (allowing perturbation on both matrices) is defined as

$$\Lambda_\epsilon(\mathbf{A}, \mathbf{B}) = \left\{ z \in \mathbb{C} : \exists \mathbf{u} \neq \mathbf{0}, (\mathbf{A}+\mathbf{E}) \cdot \mathbf{u} = z \cdot (\mathbf{B}+\mathbf{F}) \cdot \mathbf{u}, \|E\|_F^2 + \|F\|_F^2 \leq \epsilon^2 \right\}. \tag{2.1}$$

As in the case of a square matrix, when we are dealing with regular matrix pencils, there exists a nonempty spectra $\Lambda(\mathbf{A}, \mathbf{B})$ and any connected region of the pseudospectra has elements of $\Lambda(\mathbf{A}, \mathbf{B})$ inside it [10]. This does not happen with

singular matrix pencils. The definition of pseudospectra is generalized for rectangular matrices and general matrix pencils in [21], considering matrices in $\mathcal{M}_{\mathbb{C}}(m,n)$ in (2.1).

The spectra of a singular matrix pencil may be empty, as well as its pseudospectra for values of ϵ small enough. As the value of ϵ is increased, the pseudospectra will eventually become nonempty. The MPA for the generalized eigenvalue problem for nonsquare matrix pencils [3] aims to find the smallest ϵ that allows a number of eigenvalues to be included in the pseudospectra of the pair $(\mathbf{A},\mathbf{B})$. It yields not only the eigenvalues, but also the exact perturbation matrices. The formulation of the MPA, also referred to as inverse pseudospectra analysis of the pencil, is as follows:

Given $\mathbf{A},\mathbf{B}\in\mathcal{M}_{\mathbb{C}}(m,n)$, find:

$$
\begin{aligned}
&\min_{\left\{\mathbf{A_0},\mathbf{B_0},\{\lambda_k,\mathbf{v}_k\}_{k=1}^{p}\right\}} \|\mathbf{A_0}-\mathbf{A}\|_F^2+\|\mathbf{B_0}-\mathbf{B}\|_F^2\\
&\text{subject to:}\\
&\qquad\left\{\begin{array}{l}(\mathbf{A_0}-\lambda_k\mathbf{B_0})\,\mathbf{v}_k=\mathbf{0}\\ \|\mathbf{v}_k\|_2=1\end{array}\right\},\ k=1,\ldots,p\\
&\qquad\{\mathbf{v}_1,\ldots,\mathbf{v}_p\}\ \text{is an independent set.}
\end{aligned}
\tag{2.2}
$$

This difficult optimization problem is shown in [4] to be equivalent to an optimization over the compact set of $p\times p$ unitary matrices. The result in [4] provides a feasible way to compute the least-squares (LS) estimation (or maximum likelihood estimation provided the errors in the elements of the matrices are independent and follow a Gaussian distribution) of the p finite eigenvalues of a perturbed rectangular pencil. From a practical point of view, the number p of finite eigenvalues (number of finite elementary divisors in the KCF) of the pair $(\mathbf{A_0},\mathbf{B_0})$ must be known beforehand and the optimization process can be complicated due to the large number of parameters.

Two particular cases that allow for a simpler formulation of the optimization problem are studied in [3]. The eigenproblem for matrices $\mathbf{A},\mathbf{B}\in\mathcal{M}_{\mathbb{C}}(m,1)$ is studied in the first place, showing that it is equivalent to a total least squares problem. Then, the problem for matrices in $\mathcal{M}_{\mathbb{C}}(m,n)$, with $n>1$, is considered, assuming that a single finite eigenvalue is known to exist ($p=1$). The simplified formulation of the minimal perturbation approach for this particular case is as follows:

Given $\mathbf{A},\mathbf{B}\in\mathcal{M}_{\mathbb{C}}(m,n)$, find:

$$
\begin{aligned}
&\min_{\{\mathbf{A_0},\mathbf{B_0},\lambda,\mathbf{v}\}} \|\mathbf{A_0}-\mathbf{A}\|_F^2+\|\mathbf{B_0}-\mathbf{B}\|_F^2\\
&\text{subject to:}\\
&\qquad(\mathbf{A_0}-\lambda\mathbf{B_0})\,\mathbf{v}=\mathbf{0},\\
&\qquad\|\mathbf{v}\|_2=1.
\end{aligned}
\tag{2.3}
$$

A method for solving (2.3) is proposed. The optimization problem

$$
\begin{aligned}
&\min_{\{\lambda,\mathbf{v}\}} \frac{\|(\mathbf{A}-\lambda\mathbf{B})\,\mathbf{v}\|_2^2}{1+|\lambda|^2}\\
&\text{subject to:}\\
&\qquad\|\mathbf{v}\|_2^2=1
\end{aligned}
\tag{2.4}
$$

is shown to be equivalent to (2.3) (both objective functions attain their conditioned minimum values at the same points), and a numerical algorithm guaranteed to converge to the local minima of $f(\lambda, \mathbf{v}) = \frac{\|(\mathbf{A}-\lambda\mathbf{B})\mathbf{v}\|_2^2}{1+|\lambda|^2}$ is given. It is also noted that if several eigenvalues exist, $f(\lambda, \mathbf{v})$ has minima at values of λ next to the eigenvalues. Different initial values may result in the optimization process converging at different local minima, and therefore, estimations of different eigenvalues. Note that, simple as this approach may be, the estimation of p finite eigenvalues ($p > 1$) with this method is based on heuristics, unlike the LS estimation presented in [4].

When analyzing pseudospectra, we study a function that gives the norm of the smallest perturbation that must be applied to in order to make λ an eigenvalue. The pseudospectra of a square matrix $\mathbf{A}$ is studied in [20] and the proposed function is the smallest singular value of $(\mathbf{A}-\lambda\mathbf{I})$, $\sigma_{\min}(\mathbf{A}-\lambda\mathbf{I})$, as only perturbations on $\mathbf{A}$ are allowed. In [10], the pseudospectra of regular matrix pencils is studied and another function is considered, as perturbation may occur on both matrices $\mathbf{A}$ and $\mathbf{B}$. In fact, the function that appears in [10] is equivalent to $f(\lambda, \mathbf{v})$, which provides a measurement of the magnitude of the perturbation that must be applied to the pair $(\mathbf{A}, \mathbf{B})$ for λ to be its finite eigenvalue.

The position of the global minimum of $f(\lambda, \mathbf{v})$ yields a LS estimation of the eigenvalue (provided the assumptions $n > 1, p = 1$ are valid and matrices $\mathbf{A}, \mathbf{B}$ are unstructured). However, when using the proposed method for finding several eigenvalues, the minimization of $f(\lambda, \mathbf{v})$ is performed for each eigenvalue on its own, without considering how the perturbation that would make a value of λ an eigenvalue would affect the other eigenvalues in the set. It is not a LS estimation of the set of eigenvalues, but nevertheless it may be accurate enough for many applications.

3. Minimal perturbation approach for structured nonsquare matrix pencils.

3.1. Introduction. We adopt the MPA for nonsquare matrix pencils $(\mathbf{A} - \lambda\mathbf{B})$ with a particular structure, namely matrix $\mathbf{A} \in \mathcal{M}_{\mathbb{C}}(m, n)$ is the bottom $m \times n$ submatrix of some matrix $\mathbf{G} \in \mathcal{M}_{\mathbb{C}}(m+1, n)$, while $\mathbf{B} \in \mathcal{M}_{\mathbb{C}}(m, n)$ is the top $m \times n$ submatrix of $\mathbf{G}$. This structure may be written in a compact way as follows:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{0} & \mathbf{I_m} \end{bmatrix} \cdot \mathbf{G}, \\ \mathbf{B} &= \begin{bmatrix} \mathbf{I_m} & \mathbf{0} \end{bmatrix} \cdot \mathbf{G}, \end{aligned} \tag{3.1}$$

where $\mathbf{I_m}$ is an $m \times m$ identity matrix.

Structure (3.1) usually stems from the special characteristics of a particular problem or application, and matrix $\mathbf{G}$ comprises the data that is subject to perturbations. In this framework, as errors appear in the elements of $\mathbf{G}$, it is more natural to take the objective function of the MPA as the norm of the perturbation applied to matrix $\mathbf{G}$, $\|\mathbf{G_o} - \mathbf{G}\|_F^2$, instead of the sum of the norms of the perturbation suffered by matrices $\mathbf{A}$ and $\mathbf{B}$, $\|\mathbf{A_o} - \mathbf{A}\|_F^2 + \|\mathbf{B_o} - \mathbf{B}\|_F^2$. These functions are slightly different, as it can be seen from (3.1) that the elements of the first and last row of the perturbation matrix $\mathbf{G_o} - \mathbf{G}$ contribute once to the latter objective function, whereas the rest of the elements contribute twice. Taking this observation into account, the MPA for a nonsquare matrix pencil that shows structure (3.1) adopts the following conditioned

optimization formulation:

$$
\begin{aligned}
&\min_{\left\{\mathbf{G_o},\{\lambda_k,\mathbf{v}_k\}_{k=1}^{p}\right\}} \|\mathbf{G_o}-\mathbf{G}\|_F^2 \\
&\text{subject to:} \\
&\left\{ \begin{array}{l} \left(\begin{bmatrix}\mathbf{0} & \mathbf{I_m}\end{bmatrix} - \lambda_k \begin{bmatrix}\mathbf{I_m} & \mathbf{0}\end{bmatrix}\right)\mathbf{G_o}\mathbf{v}_k = \mathbf{0} \\ \|\mathbf{v}_k\|_2 = 1 \end{array} \right\},\ k=1,\ldots,p \\
&\{\mathbf{v}_1,\ldots,\mathbf{v}_p\} \text{ independent set.}
\end{aligned} \tag{3.2}
$$

As it is pointed out in [3] and section 2, an easy-to-solve optimization formulation may be obtained if the scope of the problem is limited to simple cases. We assume that matrices $(\mathbf{A},\mathbf{B})$ are such that there exists one single finite eigenvalue. As has been stated previously, if the matrix pencil comes from the perturbation of another pencil with several finite eigenvalues, these will appear as local minima of the MPA function. Problem (3.2) is simplified to

$$
\begin{aligned}
&\min_{\{\mathbf{G_o},\lambda,\mathbf{v}\}} \|\mathbf{G_o}-\mathbf{G}\|_F^2 && \text{(3.3a)} \\
&\text{subject to:} \\
&\left(\begin{bmatrix}\mathbf{0} & \mathbf{I_m}\end{bmatrix} - \lambda \begin{bmatrix}\mathbf{I_m} & \mathbf{0}\end{bmatrix}\right)\mathbf{G_o}\mathbf{v} = \mathbf{0} && \text{(3.3b)} \\
&\|\mathbf{v}\|_2 = 1. && \text{(3.3c)}
\end{aligned}
$$

In this section we give a solution for problem (3.2) as a minimization over the set $\{\{\lambda_k\}_{k=1}^{p}\}$. This solution is then particularized to the case $p=1$ (only a single eigenvalue is assumed to exist) and it is shown that the corresponding constrained optimization problem (3.3) can be transformed into an unconstrained minimization of a function of λ. The objective function of this unconstrained optimization problem can be seen as a scalar function of $\mathbb{R}^2$. Nonlinear optimization algorithms like gradient-descent, conjugate-gradient or Gauss–Newton method can be used to find a local minimum, with the advantage that the reduced number of parameters lowers the complexity of the function and the number of local minima. This procedure is simpler and faster than the iterative algorithm proposed in [3], or the minimization of $f(\lambda,\mathbf{v})$ through standard optimization algorithms.

3.2. Definition of vectors and matrices. Before performing the simplification of the optimization problem, we introduce some preliminary results that will be used. Given a set of complex numbers $\{\{\lambda_k\in\mathbb{C}\}_{k=1}^{p}\}$, matrix $\mathbf{W}_{(\lambda_1,\lambda_2,\ldots,\lambda_p)}\in\mathcal{M}_{\mathbb{C}}(m+1,p)$ is defined as the Vandermonde matrix:

$$
\mathbf{W}_{(\lambda_1,\lambda_2,\ldots,\lambda_p)} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_p \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_p^2 \\ \vdots & \vdots & & \vdots \\ \lambda_1^m & \lambda_2^m & \cdots & \lambda_p^m \end{bmatrix}. \tag{3.4}
$$

In the following, when the list of arguments is clear from the context, it will be omitted for the sake of clarity. A well-known property of Vandermonde matrices is that they have full column rank as long as $\lambda_i\neq\lambda_j$, $1\leq i\neq j\leq p$, and $(m+1)\geq p$.

We also define matrix $\mathbf{D}_{(\lambda)} \in \mathcal{M}_{\mathbb{C}}(m+1, m)$ as

$$\mathbf{D}_{(\lambda)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I_m} \end{bmatrix} - \lambda^* \begin{bmatrix} \mathbf{I_m} \\ \mathbf{0} \end{bmatrix}. \tag{3.5}$$

Matrix $\mathbf{D}_{(\lambda)}$ has some relevant properties that are easy to prove:

- $\mathbf{D}_{(\lambda)}$ has full column rank m.
- Range$\{\mathbf{D}_{(\lambda)}\}$ is an m-dimensional subspace in $\mathbb{C}^{(m+1)}$.
- The left null space of $\mathbf{D}_{(\lambda)}$ is a subspace of dimension 1 spanned by the Vandermonde vector $\mathbf{W}_{(\lambda)}$. The corresponding unit-norm vector is denoted by $\mathbf{d}_\perp \in \mathbb{C}^{(m+1)}$:

$$\mathbf{d}_\perp = \frac{1}{\sqrt{\sum_{k=0}^{m} |\lambda|^{2k}}} \cdot \begin{bmatrix} 1 \\ \lambda \\ \vdots \\ \lambda^m \end{bmatrix}. \tag{3.6}$$

As $\mathbf{D}_{(\lambda)}$ has full column rank, the hermitian matrix $\mathbf{D}_{(\lambda)}^{\mathrm{H}} \cdot \mathbf{D}_{(\lambda)}$ is regular and the projection operator $\mathbf{P_D} \in \mathcal{M}_{\mathbb{C}}(m, m)$ over Range$\{\mathbf{D}_{(\lambda)}\}$ can be computed as

$$\mathbf{P_D} = \mathbf{D}_{(\lambda)} \cdot \left(\mathbf{D}_{(\lambda)}^{\mathrm{H}} \cdot \mathbf{D}_{(\lambda)}\right)^{-1} \cdot \mathbf{D}_{(\lambda)}^{\mathrm{H}} = \left(\mathbf{I} - \mathbf{d}_\perp \cdot \mathbf{d}_\perp^{\mathrm{H}}\right). \tag{3.7}$$

3.3. Equivalent optimization problems. The following lines give a solution for problem (3.2) as a minimization over the set $\{\{\lambda_k\}_{k=1}^{p}\}$, and then the constrained optimization problem (3.3) is transformed into an unconstrained minimization of a function that only depends on λ. These transformations are made in three steps that we present as theorems.

THEOREM 1. [illegible] (3.2) [illegible]

$$\min_{\{\{\lambda_k\}_{k=1}^{p}\}} \sum_{i=1}^{p} \sigma_{(n-p+i)}\left(\mathbf{P}_{\mathbf{W}}^{\perp} \cdot \mathbf{G}\right), \tag{3.8}$$

[illegible] $\mathbf{P}_{\mathbf{W}}^{\perp}$ [illegible] $\mathbf{W}$ [illegible] (3.4) [illegible] $\mathbf{G_o}$ [illegible] p [illegible] $\mathbf{G}$

$$\mathbf{G_o} = \mathbf{G} - \mathbf{P}_{\mathbf{W}}^{\perp} \cdot \mathbf{G} \cdot \mathbf{Q} \cdot \mathbf{Q}^{\mathrm{H}}, \tag{3.9}$$

[illegible] $\mathbf{Q} \in \mathcal{M}_{\mathbb{C}}(n, p)$ [illegible] $\mathbf{P}_{\mathbf{W}}^{\perp} \cdot \mathbf{G}$ [illegible]

We will begin the transformation of the optimization problem (3.2) by noting that conditions

$$\begin{bmatrix} \mathbf{0} & \mathbf{I_m} \end{bmatrix} \cdot \mathbf{G_o} \cdot \mathbf{v}_k = \lambda_k \cdot \begin{bmatrix} \mathbf{I_m} & \mathbf{0} \end{bmatrix} \cdot \mathbf{G_o} \mathbf{v}_k, \qquad k = 1, \ldots, p \tag{3.10}$$

are equivalent to

$$\begin{cases} [\mathbf{G_o} \cdot \mathbf{v}_k]_1 = h_k, \\ [\mathbf{G_o} \cdot \mathbf{v}_k]_i = \lambda_k \cdot [\mathbf{G_o} \cdot \mathbf{v}_k]_{(i-1)} \quad i = 2, \ldots, m+1, \end{cases} \qquad k = 1, \ldots, p$$

where $[\mathbf{x}]_i$ denotes the ith element of vector $\mathbf{x}$, and $h_k \in \mathbb{C}$, $k = 1, \ldots, p$. Therefore, (3.10) will hold if and only if

$$\mathbf{G_o} \cdot \mathbf{V} = \mathbf{W} \cdot \mathbf{H} \tag{3.11}$$

for some diagonal matrix $\mathbf{H} \in \mathcal{M}_{\mathbb{C}}(p,p)$ built with $h_1, h_2, \ldots, h_p$ as its diagonal elements. Since the Vandermonde matrix $\mathbf{W} \in \mathcal{M}_{\mathbb{C}}(m+1,p)$ has full column rank (3.4), so does $\mathbf{V}$, which amounts to the set $\{\{\mathbf{v}_k\}_{k=1}^{p}\}$ being independent, as required by the last condition of (3.2).

Let $\mathbf{T} \in \mathcal{M}_{\mathbb{C}}(p,p)$ be an invertible matrix such that $\mathbf{V} \cdot \mathbf{T} = \mathbf{Q}$, with $\mathbf{Q} \in \mathcal{M}_{\mathbb{C}}(n,p)$ a matrix with orthonormal columns. Then, multiplying (3.11) by $\mathbf{T}$ we get

$$\mathbf{G_o} \cdot \mathbf{V} \cdot \mathbf{T} = \mathbf{W} \cdot \mathbf{H} \cdot \mathbf{T},$$

or, equivalently,

$$\mathbf{G_o} \cdot \mathbf{Q} = \mathbf{W} \cdot \mathbf{R}. \tag{3.12}$$

Equation (3.12) is equivalent to equation (3.10) and $\mathbf{R} \in \mathcal{M}_{\mathbb{C}}(p,p)$ is a regular matrix.

Let $\mathbf{U} \in \mathcal{M}_{\mathbb{C}}(n, n-p)$ denote a matrix with orthonormal columns such that $\mathbf{S} = \begin{bmatrix} \mathbf{Q} & \mathbf{U} \end{bmatrix}$ is a $n \times n$ unitary matrix. Taking $\mathbf{\Delta} = \mathbf{G_o} - \mathbf{G}$, the objective function of the optimization problem can be written as follows:

$$\begin{aligned} \|\mathbf{G} - \mathbf{G_o}\|_F = \|\mathbf{\Delta}\|_F = \|\mathbf{\Delta} \cdot \mathbf{S}\|_F = \left\| \begin{bmatrix} \mathbf{\Delta} \cdot \mathbf{Q} & \mathbf{\Delta} \cdot \mathbf{U} \end{bmatrix} \right\|_F \\ = \left\| \begin{bmatrix} \mathbf{\Delta}^{(\mathbf{Q})} \cdot \mathbf{Q} & \mathbf{\Delta}^{(\mathbf{U})} \cdot \mathbf{U} \end{bmatrix} \right\|_F . \end{aligned} \tag{3.13}$$

In the last equality, the decomposition of the perturbation matrix $\mathbf{\Delta} = \mathbf{\Delta}^{(\mathbf{Q})} + \mathbf{\Delta}^{(\mathbf{U})}$ has been used:

$$\mathbf{\Delta} = \mathbf{\Delta} \cdot \mathbf{S} \cdot \mathbf{S}^{\mathrm{H}} = \mathbf{\Delta} \cdot \left(\mathbf{Q} \cdot \mathbf{Q}^{\mathrm{H}} + \mathbf{U} \cdot \mathbf{U}^{\mathrm{H}} \right) = \mathbf{\Delta} \cdot \mathbf{Q} \cdot \mathbf{Q}^{\mathrm{H}} + \mathbf{\Delta} \cdot \mathbf{U} \cdot \mathbf{U}^{\mathrm{H}} = \mathbf{\Delta}^{(\mathbf{Q})} + \mathbf{\Delta}^{(\mathbf{U})}.$$

The rows of $\mathbf{\Delta}^{(\mathbf{Q})}$ and $\mathbf{\Delta}^{(\mathbf{U})}$ are the projections of the rows of $\mathbf{\Delta}$ onto the range of $\mathbf{Q}$ and $\mathbf{U}$, respectively, so $\mathbf{\Delta}^{(\mathbf{Q})} \cdot \mathbf{U} = \mathbf{0}$ and $\mathbf{\Delta}^{(\mathbf{U})} \cdot \mathbf{Q} = \mathbf{0}$.

Then, condition (3.12) can be written in terms of the perturbation matrix as

$$\left(\mathbf{G} + \mathbf{\Delta}^{(\mathbf{Q})} \right) \cdot \mathbf{Q} = \mathbf{W} \cdot \mathbf{R}. \tag{3.14}$$

It is clear that the minimum of the objective function (3.13) subject to condition (3.14) will be attained for a perturbation matrix such that $\mathbf{\Delta}^{(\mathbf{U})} = \mathbf{0}$. The objective function can be replaced by $\left\| \mathbf{\Delta}^{(\mathbf{Q})} \cdot \mathbf{Q} \right\|_F$ and, consequently, problem (3.3) is equivalent to

$$\min_{\left\{ \mathbf{G_o}, \{\lambda\}_{k=1}^{p}, \mathbf{Q} \right\}} \|\mathbf{G} \cdot \mathbf{Q} - \mathbf{G_o} \cdot \mathbf{Q}\|_F \tag{3.15a}$$

subject to:

$$\mathbf{G_o} \cdot \mathbf{Q} = \mathbf{W} \cdot \mathbf{R} \tag{3.15b}$$

for some invertible matrix $\mathbf{R} \in \mathcal{M}_{\mathbb{C}}(p,p)$ and with $\mathbf{Q} \in \mathcal{M}_{\mathbb{C}}(n,p)$ a matrix with orthonormal columns. The matrix $\mathbf{G_o}$ that solves the constrained optimization problem satisfies

$$\mathbf{G_o} = \mathbf{G} + \mathbf{\Delta} = \mathbf{G} + \mathbf{\Delta}^{(\mathbf{Q})} = \mathbf{G} + \mathbf{\Delta}^{(\mathbf{Q})} \cdot \mathbf{Q} \cdot \mathbf{Q}^{\mathrm{H}}.$$

From condition (3.14), we have

$$\mathbf{G_o} = \mathbf{G} + \mathbf{W}\cdot\mathbf{R}\cdot\mathbf{Q}^{\mathrm{H}} - \mathbf{G}\cdot\mathbf{Q}\cdot\mathbf{Q}^{\mathrm{H}}. \tag{3.16}$$

Condition (3.15b) can be incorporated to the objective function (3.15a), yielding the following optimization problem:

$$\min_{\left\{\mathbf{R},\{\lambda_k\}_{k=1}^{p},\mathbf{Q}\right\}} \left\|\mathbf{G}\cdot\mathbf{Q} - \mathbf{W}\cdot\mathbf{R}\right\|_F. \tag{3.17}$$

Given matrix $\mathbf{Q}$ and a set $\{\lambda_k\}_{k=1}^{p}$, the value of $\mathbf{R}$ that minimizes the objective function (3.17) is the one that makes the columns of $\mathbf{W}\cdot\mathbf{R}$ equal to the projection of the corresponding columns of $\mathbf{G}\cdot\mathbf{Q}$ over the range of $\mathbf{W}$:

$$\mathbf{W}\cdot\mathbf{R} = \mathbf{P_W}\cdot\mathbf{G}\cdot\mathbf{Q}. \tag{3.18}$$

Hence, the objective function can be written as:

$$\left\|\mathbf{G}\cdot\mathbf{Q} - \mathbf{P_W}\cdot\mathbf{G}\cdot\mathbf{Q}\right\|_F = \left\|(\mathbf{I}-\mathbf{P_W})\cdot\mathbf{G}\cdot\mathbf{Q}\right\|_F = \left\|\mathbf{P}_{\mathbf{W}}^{\perp}\cdot\mathbf{G}\cdot\mathbf{Q}\right\|_F.$$

The minimum of the objective function will be attained for a matrix $\mathbf{Q}$ such that its columns are the singular vectors of $\mathbf{P}_{\mathbf{W}}^{\perp}\cdot\mathbf{G}$ corresponding to its smallest singular values:

$$\min_{\left\{\{\lambda_k\}_{k=1}^{p},\mathbf{Q}\right\}} \left\|\mathbf{P}_{\mathbf{W}}^{\perp}\cdot\mathbf{G}\cdot\mathbf{Q}\right\|_F = \min_{\left\{\{\lambda_k\}_{k=1}^{p}\right\}} \sum_{i=1}^{p} \sigma_{(n-p+i)}\left(\mathbf{P}_{\mathbf{W}}^{\perp}\cdot\mathbf{G}\right).$$

Finally, an expression for the optimal solution $\mathbf{G_o}$ in terms of matrix $\mathbf{G}$ and a perturbation will be given. From (3.16) and (3.18), we have

$$\begin{aligned}\mathbf{G_o} &= \mathbf{G} + \mathbf{P_W}\cdot\mathbf{G}\cdot\mathbf{Q}\cdot\mathbf{Q}^{\mathrm{H}} - \mathbf{G}\cdot\mathbf{Q}\cdot\mathbf{Q}^{\mathrm{H}} \\ &= \mathbf{G} - (\mathbf{I}-\mathbf{P_W})\cdot\mathbf{G}\cdot\mathbf{Q}\cdot\mathbf{Q}^{\mathrm{H}} = \mathbf{G} - \mathbf{P}_{\mathbf{W}}^{\perp}\cdot\mathbf{G}\cdot\mathbf{Q}\cdot\mathbf{Q}^{\mathrm{H}}. \qquad \square\end{aligned}$$

COROLLARY 1. [illegible] (3.3) [illegible]

$$\min_{\{\lambda,\mathbf{v}\}} \mathbf{v}^{\mathrm{H}}\cdot\mathbf{G}^{\mathrm{H}}\cdot\mathbf{P_D}\cdot\mathbf{G}\cdot\mathbf{v} \tag{3.19a}$$

[illegible]

$$\|\mathbf{v}\|_2 = 1 \tag{3.19b}$$

[illegible] $\mathbf{P_D}$ [illegible] (3.7) [illegible] $\mathbf{G_o}$ [illegible] $\mathbf{G}$

$$\mathbf{G_o} = \mathbf{G} - \mathbf{P_D}\cdot\mathbf{G}\cdot\mathbf{v}\cdot\mathbf{v}^{\mathrm{H}}.$$

[illegible]. This corollary follows directly from Theorem 1, taking $p = 1$, and the definitions in section 3.2. Note that in this case $\mathbf{W} = h_1\cdot\mathbf{d}_{\perp}$, for some $h_1 \in \mathbb{C}$, and $\mathbf{Q} = \mathbf{v}$. $\square$

Corollary 1 provides a simple formulation for the MPA when only one eigenvalue is assumed to exist. Still, the minimization problem it poses is complicated due to the number of parameters involved, which discourages the use of standard optimization techniques. The following theorems show that if matrix $\mathbf{G}$ has orthonormal columns,

then problem (3.8) is equivalent to the unconstrained minimization of a function of λ.

The assumption of $\mathbf{G}$ having orthonormal columns is valid in many practical applications. For example, in the case of the greatest common divisor (GCD) computation, the columns of matrix $\mathbf{G}$ span the null-space of a matrix $\mathbf{P}$ defined in terms of the coefficients of the polynomials [14], so $\mathbf{G}$ can be computed with orthonormal columns from the singular value decomposition of $\mathbf{P}$.

On the other hand, the generalized eigenvalues of a matrix pencil that shows structure (3.1) are the roots of the Vandermonde vectors $\begin{bmatrix}1 & \lambda^* & \dots & (\lambda^*)^m\end{bmatrix}^{\mathrm{H}}$ that are included in Range($\mathbf{G}$) [18]. Therefore, a matrix $\widehat{\mathbf{G}}$ with orthonormal columns that has the same range as $\mathbf{G}$ gives rise to a matrix pencil $(\widehat{\mathbf{A}}-\lambda\cdot\widehat{\mathbf{B}})$, with $\widehat{\mathbf{A}}=\begin{bmatrix}\mathbf{0} & \mathbf{I_m}\end{bmatrix}\cdot\widehat{\mathbf{G}}$ and $\widehat{\mathbf{B}}=\begin{bmatrix}\mathbf{I_m} & \mathbf{0}\end{bmatrix}\cdot\widehat{\mathbf{G}}$, that has the same eigenvalues.

Although the MPA yields different results for $\mathbf{G}$ and $\widehat{\mathbf{G}}$, the position of the minima of the objective function when $\widehat{\mathbf{G}}$ is considered may still be a good estimate for the approximate eigenvalues of $(\mathbf{A}-\lambda\mathbf{B})$.

THEOREM 2. [illegible] (3.19) [illegible]

(3.20a) $$\max_{\{\lambda,\mathbf{v}\}} \left|\mathbf{d}_\perp^{\mathrm{H}}\cdot\mathbf{G}\cdot\mathbf{v}\right|^2$$

[illegible]

(3.20b) $$\mathbf{e}_\perp^{\mathrm{H}}\cdot\mathbf{G}\cdot\mathbf{v}=0.$$

(3.20c) $$\|\mathbf{v}\|_2=1.$$

[illegible] We will make use of the inner-product and projection function notation for transforming the objective function (3.19a):

$$\begin{aligned}\mathbf{v}^{\mathrm{H}}\cdot\mathbf{G}^{\mathrm{H}}\cdot\mathbf{P}_{\mathbf{D}}\cdot\mathbf{G}\cdot\mathbf{v} &= \left\langle \mathbf{G}\cdot\mathbf{v}, P_{\mathbf{D}_{(\lambda)}}(\mathbf{G}\cdot\mathbf{v})\right\rangle \\ &= \left\langle P_{\mathbf{D}_{(\lambda)}}(\mathbf{G}\cdot\mathbf{v})+P_{\mathbf{d}_\perp}(\mathbf{G}\cdot\mathbf{v}), P_{\mathbf{D}_{(\lambda)}}(\mathbf{G}\cdot\mathbf{v})\right\rangle \\ &= \left\langle P_{\mathbf{D}_{(\lambda)}}(\mathbf{G}\cdot\mathbf{v}), P_{\mathbf{D}_{(\lambda)}}(\mathbf{G}\cdot\mathbf{v})\right\rangle, \end{aligned} \tag{3.21}$$

where the orthogonality of projections onto the range of $\mathbf{D}_{(\lambda)}$ and $\mathbf{d}_\perp$ has been taken into account.

Under the assumption that the columns of $\mathbf{G}$ are orthonormal, the norm of $\mathbf{G}\cdot\mathbf{v}$ will be the same as the norm of $\mathbf{v}$, which must be equal to 1 (3.19b). This property fixes a relationship between the inner-products of the projections of $\mathbf{G}\cdot\mathbf{v}$:

$$\begin{aligned}\|\mathbf{G}\cdot\mathbf{v}\|_2^2 &= \langle \mathbf{G}\cdot\mathbf{v}, \mathbf{G}\cdot\mathbf{v}\rangle \\ &= \left\langle P_{\mathbf{D}_{(\lambda)}}(\mathbf{G}\cdot\mathbf{v})+P_{\mathbf{d}_\perp}(\mathbf{G}\cdot\mathbf{v}), P_{\mathbf{D}_{(\lambda)}}(\mathbf{G}\cdot\mathbf{v})+P_{\mathbf{d}_\perp}(\mathbf{G}\cdot\mathbf{v})\right\rangle \\ &= \left\langle P_{\mathbf{D}_{(\lambda)}}(\mathbf{G}\cdot\mathbf{v}), P_{\mathbf{D}_{(\lambda)}}(\mathbf{G}\cdot\mathbf{v})\right\rangle + \langle P_{\mathbf{d}_\perp}(\mathbf{G}\cdot\mathbf{v}), P_{\mathbf{d}_\perp}(\mathbf{G}\cdot\mathbf{v})\rangle = 1. \end{aligned} \tag{3.22}$$

From (3.21) and (3.22) it is clear that our objective function is

$$\|\mathbf{G}-\mathbf{G_o}\|_F^2 = \|\mathbf{G}\cdot\mathbf{v}\|_2^2 - \|P_{\mathbf{d}_\perp}(\mathbf{G}\cdot\mathbf{v})\|_2^2 = 1-\|P_{\mathbf{d}_\perp}(\mathbf{G}\cdot\mathbf{v})\|_2^2.$$

Its minimization is equivalent to the maximization of $\|P_{\mathbf{d}_\perp}(\mathbf{G}\cdot\mathbf{v})\|_2^2$.

Consequently, the objective of our optimization problem is the maximization of

$$\|P_{\mathbf{d}_\perp}(\mathbf{G}\cdot\mathbf{v})\|_2^2 = \left\|\mathbf{d}_\perp\cdot\mathbf{d}_\perp^{\mathrm{H}}\cdot\mathbf{G}\cdot\mathbf{v}\right\|_2^2 = \|\mathbf{d}_\perp\|_2^2\cdot\left|\mathbf{d}_\perp^{\mathrm{H}}\cdot\mathbf{G}\cdot\mathbf{v}\right|^2 = \left|\mathbf{d}_\perp^{\mathrm{H}}\cdot\mathbf{G}\cdot\mathbf{v}\right|^2. \qquad \square$$

THEOREM 3. [illegible] (3.20) [illegible]

$$\min_{\lambda} \frac{1}{f(\lambda)} \tag{3.23}$$

[illegible]

$$f(\lambda) = \mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G} \cdot \mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}.$$

[illegible] We define vector $\mathbf{x}$ as the vector in the direction of $\mathbf{v}$ that makes $\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G} \cdot \mathbf{x} = 1$. Since $\mathbf{v}$ is a unit-norm vector (3.20c), $\mathbf{v} = \frac{\mathbf{x}}{\|x\|_2}$, and the objective function (3.20a) can be written in terms of $\mathbf{x}$ as

$$\left|\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G} \cdot \mathbf{v}\right|^2 = \frac{\left|\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G} \cdot \mathbf{x}\right|^2}{\|\mathbf{x}\|_2^2} = \frac{1}{\|\mathbf{x}\|_2^2}.$$

Thus, the optimization problem (3.20) is equivalent to

$$\min_{\{\lambda, \mathbf{x}\}} \|\mathbf{x}\|_2^2 \tag{3.24a}$$

subject to:

$$\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G} \cdot \mathbf{x} = 1. \tag{3.24b}$$

Condition (3.24b) poses an undetermined linear system. The minimum norm solution for $\mathbf{x}$ will minimize function (3.24a). For every λ, the minimum norm solution of the respective undetermined linear system can be computed as the product of the Moore–Penrose pseudoinverse of the coefficients matrix and the independent term. The pseudoinverse of $\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G}$ can be constructed from its singular value decomposition [12], $\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G} = u \cdot \sigma \cdot \mathbf{v}^{\mathrm{H}}$, with

$$u = 1 \qquad \sigma = \left\|\mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}\right\|_2 \qquad \mathbf{v} = \frac{\mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}}{\left\|\mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}\right\|_2}.$$

Then, the pseudoinverse of $\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G}$ is defined as

$$\left(\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G}\right)^{+} = \mathbf{v} \cdot \frac{1}{\sigma} \cdot u^{\mathrm{H}} = \frac{\mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}}{\left\|\mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}\right\|_2^2}.$$

And the minimum norm solution of the undetermined linear system is

$$\mathbf{x} = \left(\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G}\right)^{+} \cdot 1 = \left(\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G}\right)^{+}.$$

With this result, the objective function we are trying to minimize is equivalent to

$$\begin{aligned} \|\mathbf{x}\|_2^2 = \mathbf{x}^{\mathrm{H}} \cdot \mathbf{x} &= \frac{\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G} \cdot \mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}}{\left\|\mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}\right\|_2^4} = \frac{\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G} \cdot \mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}}{\left\|\mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}\right\|_2^4} \\ &= \frac{\left\|\mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}\right\|_2^2}{\left\|\mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}\right\|_2^4} = \frac{1}{\left\|\mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}\right\|_2^2} \\ &= \frac{1}{\mathbf{d}_{\perp}^{\mathrm{H}} \cdot \mathbf{G} \cdot \mathbf{G}^{\mathrm{H}} \cdot \mathbf{d}_{\perp}}. \qquad \square \end{aligned} \tag{3.25}$$

4. Example. In order to obtain a matrix pencil with the structure considered in this paper, we pose a numerical GCD computation problem. First, we take a polynomial $p(x)$ with roots -0.5, -1.25, $0.2 \pm 1.4i$ and 0.625:

$$p(x) = (x + 0.5)\cdot(x + 1.25)\cdot\left(x^2 - 0.4x + 2\right)\cdot(x - 0.625)\,.$$

Then we randomly generate 20 polynomials of degree 44 with their coefficients following a normal distribution with zero mean and variance equal to one. After multiplying these polynomials with $p(x)$, we have 20 polynomials whose GCD is $p(x)$. Matrix $\mathbf{X} \in \mathcal{M}_{\mathbb{R}}(50, 20)$ is built with the coefficients of these 20 polynomials as columns. In [14, 17] it is proved that the only generalized eigenvalues of the matrix pencil $\mathbf{A} - \lambda\mathbf{B}$ with $\mathbf{A} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\cdot\mathbf{G}$, $\mathbf{B} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\cdot\mathbf{G}$, and $\mathbf{G} \in \mathcal{M}_{\mathbb{R}}(50, 30)$ the orthogonal complement of $\mathbf{X}$, are the roots of the GCD, $p(x)$. Figure 4.1(a) shows a plot of $\epsilon(\lambda) = \log(1 - f(\lambda))$. This function has been chosen so that its minima can be clearly spotted in Figure 4.1. Five minima, located at values of λ equal to the roots of $p(x)$, can be seen.

The MPA is an alternative to traditional methods (computation of the KCF or other structure-revealing forms) when the available data are contaminated by noise and the latter yield no valid results. Figures 4.1(b) and 4.1(c) show plots of function $e(\lambda)$ for matrix $\mathbf{X}$ perturbed with matrices whose elements are taken from zero-mean normal distributions of variance $\sigma = 0.5$ and $\sigma = 1$, respectively. It can be observed that the minima are not as deep as in the noiseless case (Figure 4.1(a)) and that they are slightly deviated from their correct position. Some minima may vanish for strong perturbations, and therefore, in these situations, the MPA may fail to provide an estimation of the generalized eigenvalues.

Having established the suitability of the MPA for low and moderate noise levels in our example, we proceed to compare the optimization method proposed in [3] and the minimization of function $e(\lambda)$ (3.23). To this end, we must choose a method for the unconstrained minimization of $e(\lambda)$. Due to its simplicity, we make use of the steepest descent method, although more sophisticated strategies, such as trust region methods, show better convergence properties [16].

Figure 4.1(d) shows a plot of $e(\lambda)$ for a perturbation of level $\sigma = 0.5$ with the points reached at every iteration for both minimization methods, starting from a hand-picked initial point. The steepest descent method over $e(\lambda)$ (triangles) converges to the minimum faster than the optimization algorithm proposed in [3] (circles). This can also be seen in Figure 4.2. It shows the values of $e(\lambda)$ at every iteration for two different hand-picked initial points for both optimization methods. Lines of the same style (solid or dashed) denote the same initialization, whereas triangles correspond to points found with the steepest descent method and circles correspond to the method proposed in [3]. Note that this hand-picked starting points are considerably further away from the location of the minima than the initial values of λ obtained by the squaring method of initialization [3].

Apart from the rate of convergence, another issue that should be taken into account is the numerical complexity of each iteration. The method of minimization for unstructured matrix pencils requires the computation of the singular vector related to the smallest singular value of $(\mathbf{A} - \lambda\mathbf{B})$ at every iteration. The SVD for a $m \times n$ matrix is an operation of complexity $O\left(mn^2\right)$ (this value can be lowered by suitable factorization or partial decomposition [12]). On the other hand, the gradient and Hessian of $f(\lambda)$ can be computed with a complexity of $O(m)$.

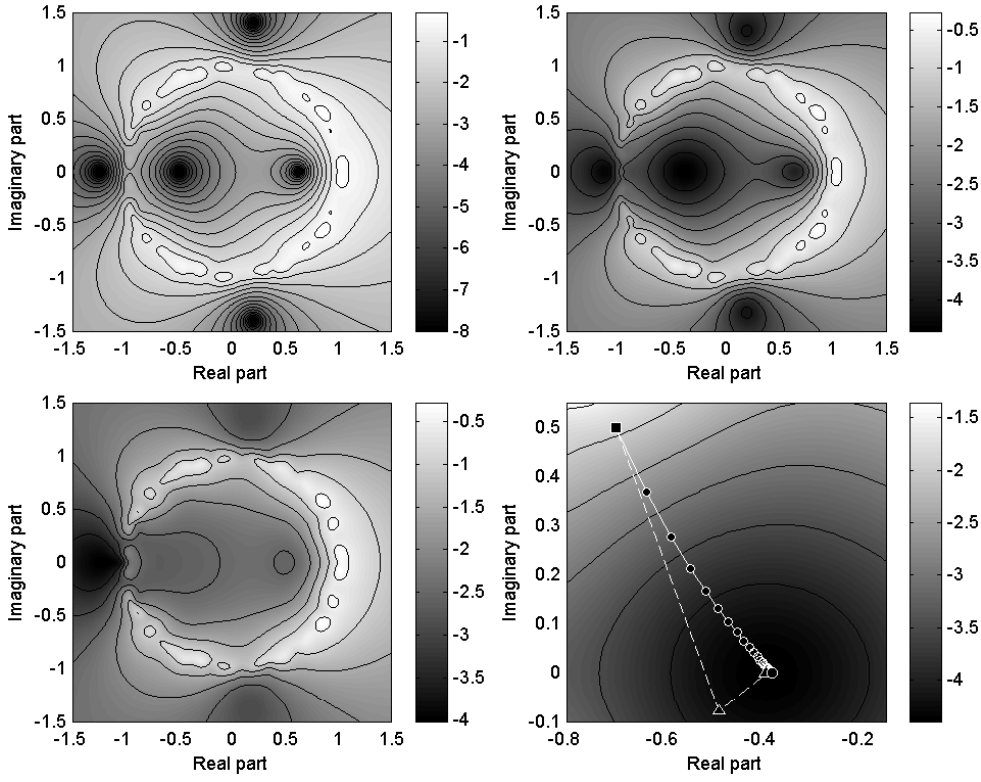


FIG. 4.1. *Log of function* $(1-f(\lambda))$ *with exact data* (a)*, with* $\sigma = 0.5$ (b) *and with* $\sigma = 1$ (c)*. Iterations of minimization algorithms* (d)*: Method proposed in* [3] *(circles, solid line) and steepest descent over* $f(\lambda)$ *defined in* (3.23) *(triangles, dashed line).*

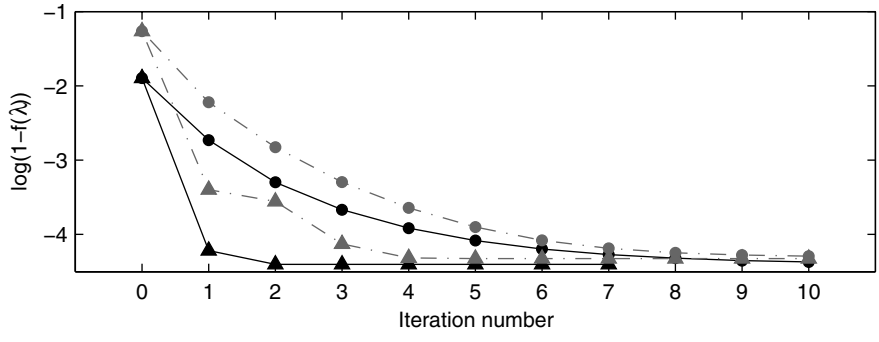


FIG. 4.2. *Values of* $\log(1-f(\lambda))$ *at every iteration for different starting points for the method proposed in* [3] *(circles) and steepest descent over* $f(\lambda)$ *defined in* (3.23) *(triangles).*

5. Conclusions and future work. This work has addressed the MPA for nonsquare matrix pencils $(\mathbf{A} - \lambda\mathbf{B})$. We have considered the following structure for matrices $\mathbf{A}, \mathbf{B} \in \mathcal{M}_{\mathbb{C}}(m, n)$:

$$\mathbf{A} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\cdot\mathbf{G}, \qquad \mathbf{B} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\cdot\mathbf{G},$$

with $\mathbf{G} \in \mathcal{M}_{\mathbb{C}}(m+1, n)$. The matrices of the pencil are derived from matrix $\mathbf{G}$. This observation leads to a slightly different MPA formulation, as the norm of the perturbation in $\mathbf{G}$ is considered, instead of the norm of the perturbations in $\mathbf{A}$ and $\mathbf{B}$. The MPA can be formulated as a minimization over the set $\{\{\lambda_k\}_{k=1}^{p}\}$. Besides, when $p = 1$, this structure allows the transformation of the constrained optimization problem that must be solved to compute the solution of the MPA into an unconstrained one. Then, well-known methods for unconstrained minimization can be used, improving the rate of convergence and simplicity of previous algorithm for the computation of the solution.

The structure considered in this paper appears in some signal processing methods and, more notably, in the numerical computation of the GCD of polynomials. However, once the GCD computation problem is formulated as an eigenvalue problem for a rectangular matrix pencil, previous proposed solutions consist in matrix transformations that reveal the generalized eigenvalues. This way of proceeding is not suitable for perturbed data, due to the ill-posedness of the transformations, while the MPA may still yield valid results in this situation. Our work in the future will focus in this application, comparing this method to other numerical GCD computation methods.

Acknowledgments. The authors would like to thank professor I. Lizasoáin for her very helpful comments and suggestions while this article was being written. The authors would also like to thank the referees for their valuable comments and suggestion greatly improving this paper.

REFERENCES

[1] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., *Templates for the Solution of Algebraic Eigenvalue Problems*, Software-Environments-Tools, SIAM, Philadelphia, 2000.

[2] D. Boley, *Estimating the sensitivity of the algebraic sctructure of pencils with simple eigenvalue estimates*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 632–643.

[3] G. Boutry, M. Elad, G. H. Golub, and P. Milanfar, *The generalized eigenvalue problem for nonsquare pencils using a minimal perturbation approach*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 582–601.

[4] D. Chu and G. H. Golub, *On a generalized eigenvalue problem for nonsquare pencils*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 770–787.

[5] J. W. Demmel and B. Kågström, *Computing stable eigendecompositions of matrix pencils*, Linear Algebra Appl., 88/89 (1987), pp. 139–186.

[6] P. M. Van Dooren, *The generalized eigenstructure problem in linear system theory*, IEEE Trans. Automat. Control, 26 (1981), pp. 111–129.

[7] J. L. M. Van Dorsselaer, *Pseudospectra for matrix pencils and stability of equilibria*, BIT, 37 (1997), pp. 833–845.

[8] A. Edelman, E. Elmroth, and B. Kågström, *A geometric approach to perturbation theory of matrices and matrix pencils. Part* I: *Versal deformations*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 653–692.

[9] M. Elad, P. Milanfar, and G. H. Golub, *Shape from moments—an estimation theory perspective*, IEEE Trans. Signal Process., 52 (2004), pp. 1814–1829.

[10] V. Frayss , M. Gueury, F. Nicoud, and V. Toumazou, *Spectral Portraits for Matrix Pencils*, Tech. Report TR/PA/96/19, CERFACS, Toulouse, France, 1996.

[11] F. R. Gantmacher, *Theory of Matrices*, Vols. 1 and 2, Chelsea, New York, 1959.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1996.

[13] Y. Hua and T. K. Sarkar, *Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise*, IEEE Trans. Acoust., Speech Signal Process., 38 (1990), pp. 814–824.

[14] N. Karcanias, *Applications of Matrix Theory*, Clarendon Press, Oxford, UK, 1989, pp. 237–248.

[15] N. Karcanias and M. Mitrouli, *A matrix pencil based numerical method for the computation of GCD of polynomials*, IEEE Trans. Automat. Control, 39 (1994), pp. 977–981.
[16] C. T. Kelley, *Iterative Methods for Optimization*, Frontiers in Applied Mathematics 18, SIAM, Philadelphia, 1999.
[17] P. Lecumberri, M. Gómez, and A. Carlosena, *Multichannel Blind Deconvolution of Transient Impulsive Signals*, in Proceedings of the 23rd IEEE Instrumentation and Measurement Technology Conference, Antalya, Turkey, 2006.
[18] P. Lecumberrri, *The GCD of Polynomials and Application to Blind Deconvolution*, Ph.D. thesis, Universidad Pública de Navarra, Pamplona, Navarra, Spain, 2007.
[19] G. W. Stewart, *Perturbation theory for rectangular matrix pencils*, Linear Algebra Appl., 208/209 (1994), pp. 297–301.
[20] N. N. Trefethen, *Computation of pseudospectra*, Acta Numerica, 8 (1999), pp. 247–295.
[21] T. G. Wright and L. N. Trefethen, *Pseudospectra of rectangular matrices*, IMA J. Numer. Anal., 22 (2002), pp. 501–519.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 56–66

FIRST-ORDER METHODS FOR SPARSE COVARIANCE SELECTION*

ALEXANDRE D'ASPREMONT†, ONUREENA BANERJEE‡, AND LAURENT EL GHAOUI‡

Abstract. Given a sample covariance matrix, we solve a maximum likelihood problem penalized by the number of nonzero coefficients in the inverse covariance matrix. Our objective is to find a sparse representation of the sample data and to highlight conditional independence relationships between the sample variables. We first formulate a convex relaxation of this combinatorial problem, we then detail two efficient first-order algorithms with low memory requirements to solve large-scale, dense problem instances.

Key words. covariance selection, semidefinite programming, coordinate descent

AMS subject classifications. 90C22, 62H20, 90C27

DOI. 10.1137/060670985

1. Introduction.

We discuss a problem of model selection.[1] Given n variables drawn from a Gaussian distribution $\mathcal{N}(0, C)$, where the true covariance matrix C is unknown, we estimate C from a sample covariance matrix Σ by maximizing its log-likelihood. Following [7], setting a certain number of coefficients in the inverse covariance matrix Σ^{-1} to zero, a procedure known as [illegible], improves the stability of this estimation procedure by reducing the number of parameters to estimate and highlight structure in the underlying model.

Here, we focus on the problem of [illegible] this pattern of zeroes in the inverse covariance matrix. We seek to trade-off the log-likelihood of the solution with the number of zeroes in its inverse, and solve the following estimation problem:

$$\begin{array}{ll} \text{maximize} & \log\det X - \langle \Sigma, X\rangle - \rho\,\mathbf{Card}(X) \\ \text{subject to} & \alpha\mathbf{I}_n \preceq X \preceq \beta\mathbf{I}_n \end{array} \tag{1.1}$$

in the variable $X \in \mathbf{S}_n$, where $\Sigma \in \mathbf{S}_n^+$ is the sample covariance matrix, $\mathbf{Card}(X)$ is the cardinality of X, i.e., the number of nonzero components in X, $\rho > 0$ is a parameter controlling the trade-off between log-likelihood and cardinality, and finally, $\alpha,\ \beta > 0$ fix bounds on the eigenvalues of the solution.

Zeroes in the inverse covariance matrix correspond to conditionally independent variables in the model and this approach can be used to simultaneously determine a robust estimate of the covariance matrix and, perhaps more importantly, discover [illegible] in the underlying graphical model. In particular, we can view (1.1) as a model selection problem using Akaike (AIC, see [1]) or Bayes (BIC, see [5]) information criterions. Both these problems can be written as in (1.1) with $\rho = 2/N$ for the AIC problem and $\rho = 2\log(N/2)/N$ for the BIC problem, where N is the sample size.

*Received by the editors September 28, 2006; accepted for publication (in revised form) by L. Vandenberghe May 31, 2007; published electronically January 25, 2008. This research was supported by NSF grant DMS-0625352, EUROCONTROL grant C20083E/BM/05, and a gift from Google, Inc.
http://www.siam.org/journals/simax/30-1/67098.html

†ORFE Department, Princeton University, Princeton, NJ 08544 (aspremon@princeton.edu).

‡EECS Department, UC Berkeley, Berkeley, CA 94720 (onureena@eecs.berkeley.edu, elghaoui@eecs.berkeley.edu).

[1]A subset of the results discussed here appeared in the Proceedings of the International Conference on Machine Learning, Pittsburgh, PA, 2006.

This has applications in speech recognition (see [2, 3]) or gene networks analysis (see [9, 8], for example).

The **Card**(X) penalty term makes the estimation problem (1.1) combinatorial (NP-hard in fact), and our first objective here is to derive a convex relaxation to this problem which can be solved efficiently. We then derive two first-order algorithms geared towards memory efficiency and large-scale, [illegible] problem instances.

In [3], Bilmes proposed a method for covariance selection based on choosing statistical dependencies according to conditional mutual information computed using training data. Other recent work involves identifying those Gaussian graphical models that are best supported by the data and any available prior information on the covariance matrix. This approach is used by [13, 9] on gene expression data. Recently, [6, 12] also considered penalized maximum likelihood estimation for covariance selection. In contrast to our results here, [12] works on the Cholesky decomposition of X using an iterative (heuristic) algorithm to minimize a nonconvex penalized likelihood problem, while [6] proposes a set of large scale interior point algorithms to solve sparse problems, i.e., problems for which the conditional independence structure is already known.

The paper is organized as follows, in section 2, we detail our convex relaxation of problem (1.1) and study the dual. In section 3, we derive two efficient algorithms to solve it. Finally, in section 4 we describe some numerical results.

2. Problem setup.

2.1. Convex relaxation. Given a sample covariance matrix $\Sigma \in \mathbf{S}_n^+$, we can write the following convex relaxation to the estimation problem (1.1):

$$\begin{array}{ll} \text{maximize} & \log\det X - \langle \Sigma, X\rangle - \rho \mathbf{1}^T|X|\mathbf{1} \\ \text{subject to} & \alpha \mathbf{I}_n \preceq X \preceq \beta \mathbf{I}_n, \end{array} \tag{2.1}$$

with variable $X \in \mathbf{S}^n$, where $\mathbf{1}$ is the n-vector of ones, so that $\mathbf{1}^T|X|\mathbf{1} = \sum_{i,j=1}^n |X_{ij}|$. The penalty term involving the sum of absolute values of the entries of X is a proxy for the number of its nonzero elements: the function $\mathbf{1}^T|X|\mathbf{1}$ can be seen as the largest convex lower bound on **Card**(X) on the hypercube, an argument used by [11] for rank minimization. It is also often used in regression techniques, such as the LASSO studied in [19], when sparsity of the solution is a concern. This relaxation is provably tight in certain cases (see [10]). In our model, the bounds (α, β) on the eigenvalues of X are fixed and user-chosen. Although we allow $\alpha = 0$ and $\beta = +\infty$, such bounds are useful in practice to control the condition number of the solution.

When $\alpha = 0$ and $\beta = +\infty$, for $\rho = 0$, provided $\Sigma \succ 0$, problems (1.1) and (2.1) have a unique solution $X^\star = \Sigma^{-1}$, and the corresponding maximum-likelihood estimate is Σ. Due to noise in the data, in practice, the sample estimate Σ may not have a sparse inverse, even if the underlying graphical model exhibits conditional independence properties. By striking a trade-off between the maximality of the likelihood and the number of nonzero elements in the inverse covariance matrix, our approach is potentially useful at [illegible], precisely conditional independence properties in the data. This means that we have to focus on the case where the matrix X is [illegible]. At the same time, it serves as a regularization technique: when Σ is rank-deficient, there is no well-defined maximum-likelihood estimate, whereas the solution to problem (2.1) is always unique and well defined for $\rho > 0$, as seen later.

2.2. Dual problem, robustness. We can rewrite the relaxation (2.1) as the following min-max problem:

$$\max_{\{X:\ \alpha\mathbf{I}_n \preceq X \preceq \beta\mathbf{I}_n\}} \min_{\{U:\ |U_{ij}|\leq\rho\}} \log\det X - \langle \Sigma + U, X\rangle, \tag{2.2}$$

which gives a natural interpretation of problem (2.1) as a worst-case [illegible] problem with componentwise bounded, additive noise on the sample covariance matrix Σ. The corresponding Lagrangian is given by

$$L(X,U,P,Q) = \log\det X - \mathbf{Tr}((\Sigma + U + Q - P)X) - \alpha\,\mathbf{Tr}\,P + \beta\,\mathbf{Tr}\,Q,$$

and we get the following dual to (2.1):

$$\begin{array}{ll} \text{minimize} & -\log\det(\Sigma + U + Q - P) - n + \alpha\,\mathbf{Tr}\,P - \beta\,\mathbf{Tr}\,Q \\ \text{subject to} & P, Q \succeq 0,\ |U_{ij}| \leq \rho, \quad i,j = 1,\ldots,n, \end{array} \tag{2.3}$$

in the variables $U, P, Q \in \mathbf{S}_n$. In what follows, we denote by $\|X\|_2$ the largest singular value of the matrix X and by $\|X\|_F$ its Frobenius norm. When $\alpha = 0$ and $\beta = +\infty$, the first-order optimality conditions impose $X(\Sigma + U) = \mathbf{I}_n$, hence we always have

$$X \succeq \alpha(n)\mathbf{I}_n \quad \text{with} \quad \alpha(n) := \frac{1}{\|\Sigma\|_2 + n\rho};$$

zero duality gap also means $\mathbf{Tr}(\Sigma X) = n - \rho\mathbf{1}^T|X|\mathbf{1}$. Because X and Σ are both positive semidefinite, we get

$$\|X\|_2 \leq \|X\|_F \leq \mathbf{1}^T|X|\mathbf{1} \leq n/\rho,$$

which, together with $\mathbf{Tr}(\Sigma X) \geq \lambda_{\min}(\Sigma)\|X\|_2$, means $\|X\|_2 \leq n/\lambda_{\min}(\Sigma)$. Finally then, we must always have

$$X \preceq \beta(n)\mathbf{I}_n \quad \text{with} \quad \beta(n) := n\min\left(\frac{1}{\rho}, \|\Sigma^{-1}\|_2\right)$$

and $0 < \alpha(n) \leq \lambda(X) \leq \beta(n) < +\infty$ at the optimum. Setting $\alpha = 0$ and $\beta = +\infty$ in problem (2.1) is then equivalent to setting $\alpha = \alpha(n)$ and $\beta = \beta(n)$. Since the objective function of problem (2.1) is strictly convex when $0 < \alpha(n) \leq \lambda(X) \leq \beta(n) < +\infty$, this shows that (2.1) always has a unique solution.

3. Algorithms. In this section, we present two algorithms for solving problem (2.1), one based on an optimal first-order method developed in [18], the other based on a block-coordinate gradient method. Of course, problem (2.1) is convex and can readily be solved using interior point methods (see [4], for example). However, such second-order methods become quickly impractical for solving (1.1), since the corresponding complexity to compute an ϵ-suboptimal solution is $O(n^6\log(1/\epsilon))$. Note, however, that we cannot expect to do better than $O(n^3)$, which is the cost of solving the nonpenalized problem for dense covariance matrices Σ.

3.1. Smooth optimization. The recently-developed first-order algorithms due to [18] trade-off a better dependence on problem size against a worst dependence on accuracy, usually $1/\epsilon$ instead of its logarithm and the method we describe next has a complexity of $O(n^{4.5}/\epsilon)$. In addition, the memory requirement of these first-order methods is much lower than that of interior-point methods, which involve forming a dense Hessian, and hence, become quickly prohibitive with a problem having $O(n^2)$ variables.

Nesterov's method. The algorithm in [18] supposes that the function to minimize conforms to a certain representation. This is the case for our problem here, so we first write (2.2) in the saddle-function format described in [18]:

$$\min_{X\in\mathcal{Q}_1} \; -\log\det X + \langle \Sigma, X\rangle + \rho \mathbf{1}^T|X|\mathbf{1} \equiv \min_{X\in\mathcal{Q}_1}\max_{U\in\mathcal{Q}_2} \; \hat{f}(X) + \langle A(X), U\rangle,$$

where we define $\hat{f}(X) = -\log\det X + \langle \Sigma, X\rangle$, $A = \rho I_{n^2}$, and

$$\mathcal{Q}_1 := \{X \in \mathcal{S}^n \;:\; \alpha \mathbf{I}_n \preceq X \preceq \beta \mathbf{I}_n\}, \quad \mathcal{Q}_2 := \{U \in \mathcal{S}^n \;:\; \|U\|_\infty \leq 1\}.$$

The adjoint of this problem, corresponding to the dual problem (2.3), is then written:

$$\max_{U\in\mathcal{Q}_2} \phi(U) \quad \text{where} \quad \phi(U) := \min_{X\in\mathcal{Q}_1} -\log\det X + \langle \Sigma + U, X\rangle. \tag{3.1}$$

When a function can be represented in this saddle function format, the method described in [18] combines two steps.

Smoothing step. By adding a strongly convex penalty to the saddle function representation of f, the algorithm first computes a smooth ϵ-approximation of f with Lipschitz continuous gradient. This can be seen as a generalized Moreau–Yosida regularization step (see [14], for example).

Optimal first-order scheme. The algorithm then applies the optimal first-order scheme for functions with Lipschitz continuous gradients detailed in [16] to the regularized function. Each iteration requires efficiently computing the regularized function value and its gradient. In all the semidefinite programming applications detailed here, this can be done extremely efficiently, with a complexity of $O(n^3)$ and memory requirements in $O(n^2)$. The method is only efficient if all these steps can be performed explicitly or at least very efficiently. As we will see below, this is the case here.

Prox-functions and strong convexity. To $\mathcal{Q}_1$ and $\mathcal{Q}_2$ we now associate norms and so-called prox-functions. For $\mathcal{Q}_1$, we use the Frobenius norm and a prox-function:

$$d_1(X) = -\log\det X + n\log\beta.$$

The function d_1 is strongly convex on $\mathcal{Q}_1$, with a convexity parameter of $\sigma_1 = 1/\beta^2$, in the sense that $\nabla^2 d_1(X)[H,H] = \mathbf{Tr}(X^{-1}HX^{-1}H) \geq \beta^{-2}\|H\|_F^2$ for every H. Furthermore, the center of the set, $X_0 := \arg\min_{X\in\mathcal{Q}_1} d_1(X)$ is $X_0 = \beta\mathbf{I}_n$ and satisfies $d_1(X_0) = 0$. With our choice, we have $D_1 := \max_{X\in\mathcal{Q}_1} d_1(X) = n\log(\beta/\alpha)$.

To $\mathcal{Q}_2$, we also associate the Frobenius norm and the prox-function $d_2(U) = \|U\|_F^2/2$. With this choice, the center U_0 of $\mathcal{Q}_2$ is $U_0 = 0$. Furthermore, the function d_2 is strongly convex on its domain with a convexity parameter with respect to the 2-norm $\sigma_2 = 1$, and we have $D_2 := \max_{U\in\mathcal{Q}_2} d_2(U) = n^2/2$.

The function $\hat{f}$ has a gradient that is Lipschitz-continuous with respect to the Frobenius norm on the set $\mathcal{Q}_1$ with Lipschitz constant $M = 1/\alpha^2$. Finally, the norm (induced by the Frobenius norm) of the operator $A = \rho I_{n^2}$ is ρ.

Smoothing. The method is based on replacing the objective of the original problem, $f(X)$, with $f_\epsilon(X)$, where $\epsilon > 0$ is the desired accuracy, and f_ϵ is a penalized function involving the prox-function d_2, defined as

$$f_\epsilon(X) := \hat{f}(X) + \max_{U\in\mathcal{Q}_2}\{\langle X, U\rangle - (\epsilon/2D_2)d_2(U)\}. \tag{3.2}$$

The above function turns out to be a smooth uniform approximation to f everywhere, with maximal error $\epsilon/2$. Furthermore, the function f_ϵ has a Lipschitz-continuous gradient with Lipschitz constant given by $L(\epsilon) := M + D_2\|A\|^2/(2\sigma_2\epsilon)$. A specific first-order algorithm detailed in [16] for smooth, constrained convex minimization is then applied to the function f_ϵ to get a convergence rate in $O(1/\epsilon)$.

Choose $\epsilon > 0$ and set $X_0 = \beta\mathbf{I}_n$, the algorithm then updates primal and dual iterates Y_k and $\hat{U}_k$ using the following steps:

1. Compute $\nabla f_\epsilon(X_k) = -X^{-1} + \Sigma + U^*(X_k)$, where $U^*(X)$ solves (3.2).
2. Find $Y_k = \mathrm{argmin}_{Y\in\mathcal{Q}_1}\{\langle\nabla f_\epsilon(X_k), Y - X_k\rangle + \frac{1}{2}L(\epsilon)\|Y - X_k\|_F^2\}$.
3. Find $Z_k = \mathrm{argmin}_{Z\in\mathcal{Q}_1}\left\{\frac{L(\epsilon)d_1(Z)}{\sigma_1} + \sum_{i=0}^k \frac{i+1}{2}(f_\epsilon(X_i) + \langle\nabla f_\epsilon(X_i), Z - X_i\rangle)\right\}$.
4. Update $X_k = \frac{2}{k+3}Z_k + \frac{k+1}{k+3}Y_k$ and $\hat{U}_k = \frac{k\hat{U}_{k-1}+2U^*(X_k)}{(k+2)}$.
5. Repeat until the duality gap is less than the target precision:

$$-\log\det Y_k + \langle\Sigma, Y_k\rangle + \rho\mathbf{1}^T|Y_k|\mathbf{1} - \phi(\hat{U}_k) \leq \epsilon.$$

The key to the method's success is that steps 1–3 and 5 can be performed explicitly and only involve an eigenvalue decomposition. Step one above computes the (smooth) function value and gradient. The second step computes the , which matches the gradient step for unconstrained problems (see [17, p. 86]). Steps three and four update an [17, p. 72] of f_μ whose minimum can be computed explicitly and gives an increasingly tight upper bound on the minimum of f_μ. We now present these steps in detail for our problem.

1 The first step requires computing the gradient of the function

$$f_\epsilon(X) = \hat{f}(X) + \max_{u\in\mathcal{Q}_2}\langle X, U\rangle - (\epsilon/2D_2)d_2(U).$$

This function can be expressed in closed form as $f_\epsilon(X) = \hat{f}(X) + \sum_{i,j}\psi_\mu(X_{ij})$, where

$$\psi_\epsilon(x) := \begin{cases} |x| - (\epsilon/4D_2) & \text{if } |x| \geq (\epsilon/2\rho D_2), \\ D_2x^2/\epsilon & \text{otherwise,} \end{cases}$$

which is simply the Moreau–Yosida regularization of the absolute value and the gradient of the function at X is

$$\nabla f_\mu(X) = -X^{-1} + \Sigma + U^*(X),$$

with

$$U^*(X) := \max(\min(X/\mu, \rho), -\rho),$$

with min. and max. understood componentwise. The cost of this step is dominated by that of computing the inverse of X, which is $O(n^3)$.

2 This step involves a problem of the form

$$T_{\mathcal{Q}_1}(X) = \arg\min_{Y\in\mathcal{Q}_1}\langle\nabla f_\epsilon(X), Y - X\rangle + \frac{1}{2}L\|Y - X\|_F^2,$$

where $X \in \mathcal{Q}_1$ is given. This problem can be reduced to one of projection on $\mathcal{Q}_1$, namely

$$\min_{Y\in\mathcal{Q}_1}\|Y - G\|_F^2,$$

where $G := X - L^{-1}\nabla f_\epsilon(X)$. Using the rotational invariance of this problem, we reduce it to a vector problem:

$$\min_\lambda \ \textstyle\sum_i (\lambda_i - \gamma_i)^2 \ : \ \alpha \leq \lambda_i \leq \beta, \quad i = 1, \ldots, n,$$

where γ is the vector of eigenvalues of G. This problem admits a simple explicit solution:

$$\lambda_i = \min(\max(\gamma_i, \alpha), \beta), \quad i = 1, \ldots, n.$$

The corresponding solution is then $Y = V^T \mathbf{diag}(\lambda) V$, where $G = V^T \mathbf{diag}(\gamma) V$ is the eigenvalue decomposition of G. The cost of this step is dominated by the cost of forming the eigenvalue decomposition of G, which is $O(n^3)$.

Step 3 The third step involves solving a problem of the form

$$Z := \arg\max_{X \in \mathcal{Q}_1} \ d_1(X) + \langle S, X \rangle, \tag{3.3}$$

where S is given. Again, due to the rotational invariance of the objective and feasible set, we can reduce the problem to a one-dimensional problem:

$$\min_\lambda \ \textstyle\sum_i \sigma_i \lambda_i - \log \lambda_i \ : \ \alpha \leq \lambda_i \leq \beta,$$

where σ contains the eigenvalues of S. This problem has a simple, explicit solution:

$$\lambda_i = \min(\max(1/\sigma_i, \alpha), \beta), \quad i = 1, \ldots, n.$$

The corresponding solution is then $Y = V^T \mathbf{diag}(\lambda) V$, where $S = V^T \mathbf{diag}(\sigma) V$ is the eigenvalue decomposition of S. Again, the cost of this step is dominated by the cost of forming the eigenvalue decomposition of S, which is $O(n^3)$.

Computing $\phi(\hat{U}_k)$ For a given matrix $\hat{U}_k$ the function ϕ in (3.1) is computed as

$$\phi(\hat{U}_k) = \min_{X \in \mathcal{Q}_1} - \log\det X + \langle \Sigma + \hat{U}_k, X \rangle.$$

This means projecting $(\Sigma + \hat{U}_k)^{-1}$ on $\mathcal{Q}_1$ only involves an eigenvalue decomposition.

Complexity estimate To summarize, for step 1, the gradient of f_ϵ is readily computed in closed form, via the computation of the inverse of X. Step 2 essentially amounts to projecting on $\mathcal{Q}_1$ and requires an eigenvalue problem to be solved; likewise for step 3. In fact, each iteration costs $O(n^3)$. The number of iterations necessary to achieve an objective with absolute accuracy less than ϵ is then given by

$$N(\epsilon) := 4\|A\| \frac{1}{\epsilon} \sqrt{\frac{D_1 D_2}{\sigma_1 \sigma_2}} + \sqrt{\frac{M D_1}{\sigma_1 \epsilon}} = \frac{\kappa \sqrt{n(\log \kappa)}}{\epsilon} (4n\alpha\rho + \sqrt{\epsilon}), \tag{3.4}$$

where $\kappa = \beta/\alpha$ bounds the solution's condition number. Thus, the overall complexity when $\rho > 0$ is in $O(n^{4.5}/\epsilon)$, as claimed.

3.2. Block-coordinate gradient methods. In this section, we focus on the particular case where $\alpha = 0$ and $\beta = +\infty$ (hence, implicitly $\alpha = \alpha(n)$ and $\beta = \beta(n)$) and derive gradient minimization algorithms that take advantage of the problem structure. We consider the following problem:

$$\max_X \ \log\det X - \langle \Sigma, X \rangle - \rho \mathbf{1}^T |X| \mathbf{1} \tag{3.5}$$

in the variable $X \in \mathbf{S}_n$, where $\rho > 0$ again controls the trade-off between log-likelihood and sparsity of the inverse covariance matrix. Its dual is given by

$$\begin{array}{ll} \text{minimize} & -\log\det(\Sigma + U) - n \\ \text{subject to} & |U_{ij}| \leq \rho, \quad i,j = 1,\ldots,n, \end{array} \tag{3.6}$$

in the variable $U \in \mathbf{S}_n$. We partition the matrices X and U in block format:

$$X = \begin{pmatrix} Z & x \\ x^T & y \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} V & u \\ u^T & w \end{pmatrix},$$

where $Z \succ 0$ and U are fixed and $x,\ u \in \mathbf{R}^{(n-1)},\ y,\ w \in \mathbf{R}$ are the variables (row and column) we are updating. We also partition the sample matrix according to the same block structure:

$$\Sigma = \begin{pmatrix} A & b \\ b^T & c \end{pmatrix},$$

where $A \in \mathbf{S}_{(n-1)},\ b \in \mathbf{R}^{(n-1)},\ c \in \mathbf{R}$. In the methods that follow, we will update only one column (and corresponding row) at a time and without loss of generality we can always assume that we are updating the last one.

The dual problem (3.6):

$$\begin{array}{ll} \text{minimize} & -\log\det(\Sigma + U) - n \\ \text{subject to} & |U_{ij}| \leq \rho, \quad i,j = 1,\ldots,n, \end{array}$$

in the variable $U \in \mathbf{S}_n$, can be written in block format as

$$\begin{array}{ll} \text{minimize} & -\log\det(A+V) - \log\big((w+c) - (b+u)^T(A+V)^{-1}(b+u)\big) - n \\ \text{subject to} & |w| \leq \rho,\ |u_i| \leq \rho, \quad i = 1,\ldots,n, \end{array}$$

in the variables $u \in \mathbf{R}^{(n-1)}$ and $w \in \mathbf{R}$ (V is fixed at each iteration). We directly get $w = \rho$ so the diagonal of the optimal solution must be $\rho\mathbf{1}$. The main step at each iteration is then a box constrained quadratic program (QP):

$$\begin{array}{ll} \text{minimize} & (b+u)^T(A+V)^{-1}(b+u) \\ \text{subject to} & |u_i| \leq \rho, \quad i = 1,\ldots,n, \end{array} \tag{3.7}$$

in the variable $u \in \mathbf{R}^{(n-1)}$. To summarize, the block coordinate descent algorithm proceeds as follows:

1. Pick the row and column to update.
2. Compute $(A+V)^{-1}$.
3. Solve the box constrained QP in (3.7).
4. Repeat until duality gap is less than precision: $\langle \Sigma, X\rangle - n + \rho\mathbf{1}^T|X|\mathbf{1} \leq \epsilon$.

At each iteration, we need to compute the inverse of the submatrix $(A+V) \in \mathbf{S}_{(n-1)}$, but we can update this inverse using the Sherman–Woodbury–Morrison formula on two rank-two updates; hence, it is only necessary to compute a full inverse at the first iteration.

Primal block coordinate descent. For a fixed Z, problem (3.5) is equivalent to

$$\begin{array}{ll} \text{maximize} & \log\left(y - x^T Z^{-1} x\right) - 2b^T x - y(c+\rho) - 2\rho\|x\|_1 \\ \text{subject to} & y - x^T Z^{-1} x > 0,\ y > 0, \end{array}$$

in the variables $x \in \mathbf{R}^{(n-1)}$, $y \in \mathbf{R}$, where $Z \succ 0$ (given) and the Schur complement constraints imply $X \succ 0$. We can solve for the optimal y explicitly and the problem in x becomes

$$\max_x -x^T Q x - 2b^T x - 2\rho\|x\|_1,$$

where $Q := (c+\rho)Z^{-1}$. Its dual is also box-constrained QP:

$$\begin{array}{ll} \text{minimize} & (b+u)^T Z (b+u) \\ \text{subject to} & \|u\|_\infty \leq \rho, \end{array}$$

in the variable $u \in \mathbf{R}^{(n-1)}$. At the optimum for this QP, we must have

$$x = -\frac{1}{(c+\rho)} Z(b+u), \quad \text{and} \quad y = \frac{1}{(c+\rho)} + \frac{1}{(c+\rho)^2}(b+u)^T Z (b+u),$$

and we iterate as above.

Dual block coordinate descent. The two block-coordinate methods detailed in this section both amount to solving a sequence of box-constrained quadratic program of the form

$$\begin{array}{ll} \text{minimize} & x^T A x + b^T x \\ \text{subject to} & \|x\|_\infty \leq \rho, \end{array} \tag{3.8}$$

in the variable $x \in \mathbf{R}^n$. The objective function has a Lipschitz continuous gradient with constant $L = 2\lambda^{\max}(A)$ on the box $\mathcal{B} = \{x \in \mathbf{R}^n : \|x\|_\infty \leq \rho\}$, where we can define a prox function $(1/2)\|x\|^2$ which is strongly convex with constant one and bounded above by $(1/2)n\rho^2$ on $\mathcal{B}$. From [16] or [18], we know that solving (3.8) up to a precision ϵ will require at most $2\rho\sqrt{n\lambda^{\max}(A)/\epsilon}$ iterations of the first-order method detailed in [16], with each iteration equivalent to a matrix-vector product and a projection on the box $\mathcal{B}$. This means that the total complexity of solving (3.8) is given by

$$O\left(\rho n^{2.5}\sqrt{\frac{\lambda^{\max}(A)}{\epsilon}}\right).$$

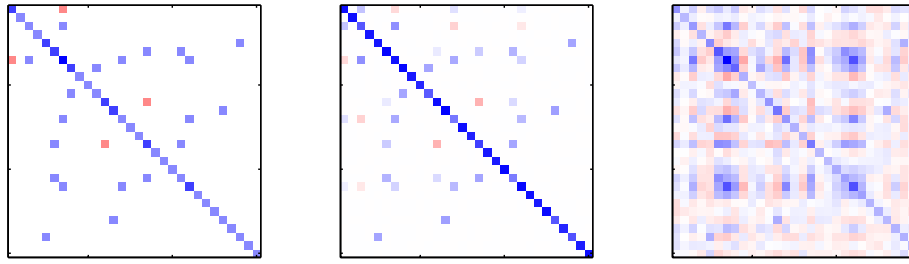
Block coordinate descent. Following [15], with block coordinate descent corresponding to coordinate descent with the almost cyclic rule (defined in [15], it simply means here that we go through each index at least once per outer iteration) and using the fact that $\log\det(X)$ satisfies the strict concavity assumptions in [15, assumption A2], we can show that the convergence rate of the block coordinate descent method is at least linear. Each iteration requires solving a box-constrained QP and takes $O(n^3 log(1/\epsilon))$ operations using an interior point solver or $O(n^{2.5}/\sqrt{\epsilon})$ using the optimal first-order scheme in [16]. We cannot use the same argument to show convergence of block coordinate ascent but empirical performance is comparable. In practice we have found that a small number of sweeps through all columns, independent of problem size n, is sufficient for convergence.

The block coordinate descent methods implemented here correspond to coordinate descent using the almost cyclic rule, alternative row/column selection rules could improve the convergence speed. Also, each iteration of the block coordinate descent method corresponds to two rank-two updates of the inverse matrix; hence, the cost of maintaining the inverse submatrix using the Sherman–Woodbury–Morrison formula is only $O(n^2)$.

4. Numerical results. In this section we test the performance of the methods detailed above on some randomly generated examples. We first form a sparse matrix A with a diagonal equal to one and a few randomly chosen, nonzero off-diagonal terms equal to $+1$ or -1. We then form the matrix

$$B = A^{-1} + \sigma V$$

where $V \in \mathbf{S}_n$ is a symmetric, i.i.d. uniform random matrix. Finally, we make B positive definite by shifting its eigenvalues, and use this noisy, random matrix to test our covariance selection methods.



Original inverse A | Solution for $\rho = 0.5$ | Noisy inverse B^{-1}

FIG. 4.1. *Recovering the sparsity pattern. We plot the original inverse covariance matrix A, the solution to problem* (2.1)*, and the noisy inverse B^{-1}.*

In Figure 4.1, we plot the sparsity patterns of the original inverse covariance matrix A, the solution to problem (2.1), and the noisy inverse B^{-1} in a randomly generated example with $n = 30$, $\sigma = 0.15$, and $\rho = 0.5$. In Figure 4.2 we represent the dependence structure of interest rates (sampled over a year) inferred from the inverse covariance matrix. Each node represents a particular interest rate maturity and the nodes are linked if the corresponding coefficient in the inverse covariance matrix is nonzero (i.e., they are conditionally dependent). We compare the solution to problem (2.1) on this matrix for $\rho = 0$ and $\rho = 0.1$ and notice that in the sparse solution the rates appear clearly clustered by maturity.

In Figure 4.3, we study computing times for various choices of algorithms and problem sizes. On the , , we plot CPU time to reduce the duality gap by a factor 10^{-2} versus problem size n, on randomly generated problems, using the coordinate descent code and the optimal first-order for solving box QPs. On the , , , , we plot duality gap versus CPU time for both smooth minimization and block-coordinate algorithms for a randomly generated problem of size $n = 250$. For the smooth minimization code, we set $\alpha = 1/\lambda^{\max}(B)$ and we plot computing time for both $\beta = 1/(2\lambda^{\min}(B))$ (smooth. opt. 1/2) and $\beta = 2/\lambda^{\min}(B)$ (smooth. opt. 2). In the examples of Figure 4.3, we notice that the numerical cost of our methods grows experimentally as $O(n^3)$.

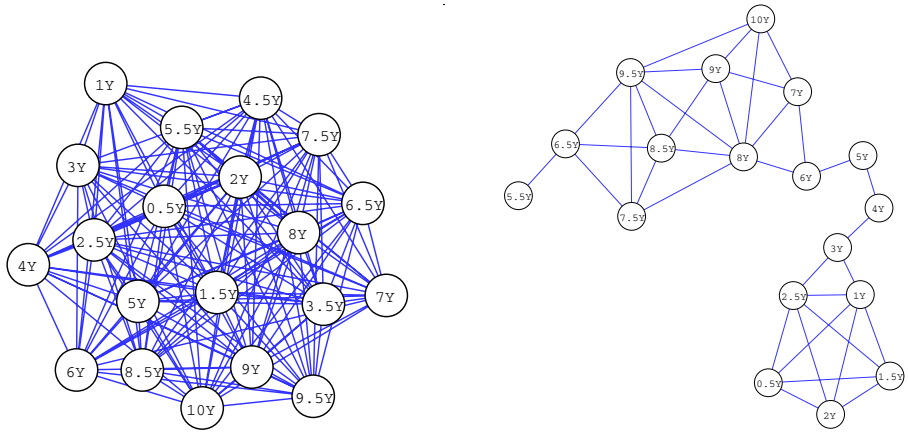


FIG. 4.2. *We plot the network formed using the solution to problem* (2.1) *on an interest rate covariance matrix for* $\rho = 0$ (left) *and* $\rho = 0.1$ (right). *In the sparse solution the rates appear clearly clustered by maturity.*

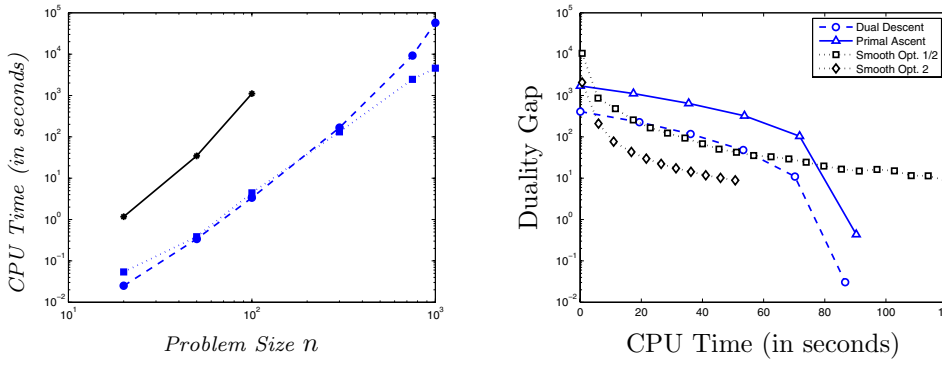


FIG. 4.3. *Computing time.* Left*: We plot CPU time to reduce the duality gap by a factor* 10^{-2} *versus problem size* n*, on randomly generated problems, using the coordinate descent code and the optimal first-order algorithm for solving box QPs (dashed line, circles), smooth minimization (dotted line, squares), and a simple conjugate gradient method with a Polak–Ribiere update, without preconditioning (solid line, stars).* Right*: We plot the duality gap versus CPU time for both smooth minimization and block-coordinate algorithms for a problem of size* $n = 250$.

Acknowledgments. The authors would like to thank Francis Bach, Peter Bartlett, and Martin Wainwright for enlightening discussions on the topic. We are grateful to the (anonymous) referees for important comments and for finding an error in an earlier version of the paper.

REFERENCES

[1] J. AKAIKE, *Information theory and an extension of the maximum likelihood principle*, in Second International Symposium on Information Theory, B. N. Petrov and F. Csaki, eds., Akedemiai Kiado, Budapest, Hungary, 1973, pp. 267–281.

[2] J. A. BILMES, *Natural Statistic Models for Automatic Speech Recognition*, Ph.D. thesis, UC Berkeley, Dept. of EECS, CS Division, Berkeley, CA, 1999.

[3] J. A. BILMES, *Factored sparse inverse covariance matrices*, IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey, 2000, pp. 1009–1012.

[4] S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, 2004.

[5] K. P. Burnham and D. R. Anderson, *Multimodel inference: Understanding AIC or BIC in model selection*, Sociol. Methods Res., 33 (2004), pp. 261–304.
[6] J. Dahl, V. Roychowdhury, and L. Vandenberghe, *Maximum Likelihood Estimation of Gaussian Graphical Models: Numerical Implementation and Topology Selection*, UCLA preprint, 2005.
[7] A. Dempster, *Covariance selection*, Biometrics, 28 (1972), pp. 157–175.
[8] A. Dobra, C. Hans, B. Jones, J. R. Nevins, G. Yao, and M. West, *Sparse graphical models for exploring gene expression data*, J. Multivariate Anal., 90 (2004), pp. 196–212.
[9] A. Dobra and M. West, *Bayesian covariance selection*, ISDS Working paper, 2004.
[10] D. L. Donoho and J. Tanner, *Sparse nonnegative solutions of underdetermined linear equations by linear programming*, Proc. Natl. Acad. Sci. USA, 102 (2005), pp. 9446–9451.
[11] M. Fazel, H. Hindi, and S. Boyd, *A rank minimization heuristic with application to minimum order system approximation*, in Proceedings of the American Control Conference, Arlington, VA, 6 (2001), pp. 4734–4739.
[12] J. Z. Huang, N. Liu, and M. Pourahmadi, *Covariance selection and estimattion via penalized normal likelihood*, Biometrika, 93 (2007), pp. 85–98.
[13] B. Jones, C. Carvalho, A. Dobra, C. Hans, C. Carter, and M. West, *Experiments in stochastic computation for high-dimensional graphical models*, Statist. Sci., 20 (2005), pp. 388–400.
[14] C. Lemaréchal and C. Sagastizábal, *Practical aspects of the Moreau-Yosida regularization: Theoretical preliminaries*, SIAM J. Optim., 7 (1997), pp. 367–385.
[15] Z. Q. Luo and P. Tseng, *On the convergence of the coordinate descent method for convex differentiable minimization*, J. Optim. Theory Appl., 72 (1992), pp. 7–35.
[16] Y. Nesterov, *A method of solving a convex programming problem with convergence rate* $O(1/k^2)$, Soviet Math. Dokl., 27 (1983), pp. 372–376.
[17] Y. Nesterov, *Introductory Lectures on Convex Optimization*, Springer, 2003.
[18] Y. Nesterov, *Smooth minimization of nonsmooth functions*, Math. Program., Ser. A, 103 (2005), pp. 127–152.
[19] R. Tibshirani, *Regression shrinkage and selection via the LASSO*, J. Roy. Statist. Soc., Ser. B, 58 (1996), pp. 267–288.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 67–83

WAVEFRONT RECONSTRUCTION METHODS FOR ADAPTIVE OPTICS SYSTEMS ON GROUND-BASED TELESCOPES*

JOHNATHAN M. BARDSLEY†

Abstract. The earth's atmosphere is not a perfect media through which to view objects in outer-space; turbulence in the atmospheric temperature distribution results in refractive index variations that interfere with the propagation of light. As a result, wavefronts are nonplanar when they reach the ground. The deviation from planarity of a wavefront is known as phase error, and it is phase error that causes the refractive blurring of images. Adaptive optics systems seek to remove phase error from incoming wavefronts. In ground-based astronomy, an estimate of the phase error in a wavefront is typically obtained from wavefront gradient measurements collected by a Shack–Hartmann sensor. The estimate is then used to create a counter wavefront, e.g., using a deformable mirror that (approximately) removes the phase error from the incoming wavefronts. The problem of reconstructing the phase error from Shack–Hartmann gradient measurements requires the solution of a large linear system whose form is defined by the configuration of the sensor. We derive this system and present both the regular least squares and minimum variance approaches to its solution. The most effective existing approaches are then presented alongside new computational methods, and comparisons are made.

Key words. adaptive optics, wavefront reconstruction, minimum variance estimation

AMS subject classifications. 65F05, 65F10

DOI. 10.1137/06067506X

1. Introduction.

The standard mathematical model for image formation in ground-based astronomy is

$$d(x,y) = \int_{\mathbb{R}^2} k(x,y;\xi,\eta) f(\xi,\eta)\, d\xi d\eta. \tag{1.1}$$

Here f is the object being viewed; d is the image of f seen by the telescope; and k is the [illegible] (PSF), which characterizes the blurring effects of the imaging system. In traditional approaches, the PSF characterizes the diffractive blur of the telescope as well as the refractive blur of the atmosphere. Adaptive optics systems, however, seek to remove the refractive effects of the atmosphere prior to image formation. If this is done exactly, so-called [illegible] is obtained, which maximizes angular resolution and sensitivity.

The idea behind adaptive optics can be illustrated using the spatially invariant PSF model

$$k[\phi](x,y) = \left| \mathcal{F}^{-1} \left\{ P(x,y) e^{i\phi(x,y)} \right\} \right|^2, \tag{1.2}$$

which is obtained using techniques from Fourier optics [5]. Here $P(x,y)$ is the telescope's pupil function, and $\phi(x,y)$ denotes the phase error, or simply the phase, and

*Received by the editors November 16, 2006; accepted for publication (in revised form) by J. G. Nagy June 15, 2007; published electronically February 6, 2008. This work was done during the author's visit to the University of Helsinki, Finland in 2006-2007 under the University of Montana Faculty Exchange Program. This work was partially supported by the NSF under grant DMS-0504325.

http://www.siam.org/journals/simax/30-1/67506.html

†Department of Mathematical Sciences, University of Montana, Missoula, MT 59812 (bardsleyj@mso.umt.edu).

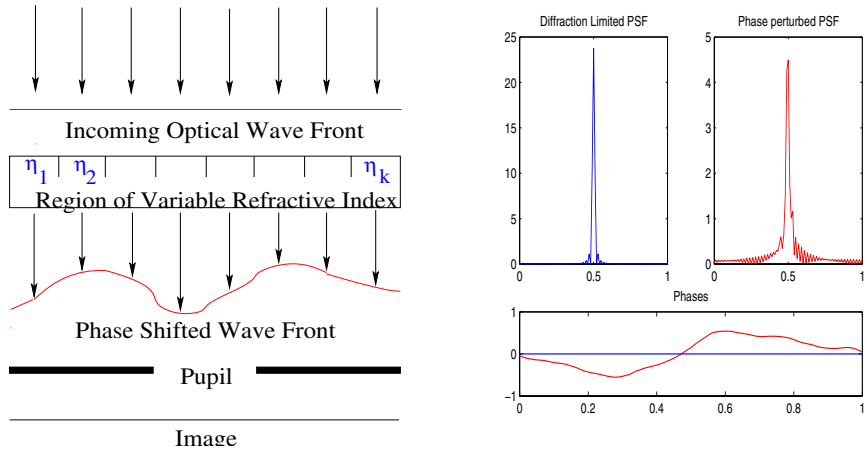


FIG. 1.1. *One-dimensional phase error schematic. On the left, the effects of refractive index variations in the earth's atmosphere are illustrated. On the right, where both diffraction limited and phase perturbed PSFs are plotted, the effects of phase errors on the corresponding PSFs are demonstrated.*

is defined to be the deviation from planarity of the wavefront at the point (x, y). For simulation and analysis, we will assume that $P(x, y)$ is 1 inside the pupil and 0 otherwise, though in practice this will not always be the case as a telescope's mirror can get dirty.

Adaptive optics systems seek to remove the phase error ϕ from the incoming wavefronts. If done exactly, the resulting PSF then has the form

$$k[0](x, y) = \left|\mathcal{F}^{-1}\left\{P(x, y)\right\}\right|^2, \tag{1.3}$$

in which case the [illegible]

$$d_{\mathrm{DL}}(x, y) \stackrel{\mathrm{def}}{=} \int_{\mathbb{R}^2} k[0](x - \xi, y - \eta) f(\xi, \eta)\, d\xi d\eta \tag{1.4}$$

is what is seen by the telescope.

Phase errors arise due to index of refraction variations in the atmosphere. A one-dimensional schematic of this process is given on the left in Figure 1.1. Since the refractive index—denoted by η_i in the schematic—determines the speed of propagation of the wavefront, variations in the refractive index result in wavefront perturbations or phase errors. To see the effects of phase errors on the PSF, on the right-hand side in Figure 1.1 we plot two PSFs—one when phase errors characteristic of atmospheric turbulence are present, and one when the phase error is zero.

The phase can be estimated in a number of ways [2]. The most common approach in ground-based astronomy is to use the Shack–Hartmann sensor, which collects measurements of the gradient of the incoming wavefronts and then seeks to reconstruct the phase from those measurements. The phase estimate is then used to create a counter wavefront distortion ϕ_{DM} via the deformation of an optical component known as a deformable mirror (DM). If the PSF has the form (1.2), then the phase corrected PSF will have the form

$$k[\phi + \phi_{\mathrm{DM}}](x, y) = \left|\mathcal{F}^{-1}\left\{P(x, y) e^{i(\phi + \phi_{\mathrm{DM}})(x, y)}\right\}\right|^2. \tag{1.5}$$

Ideally, the DM created counter wavefront satisfies $\phi_{\mathrm{DM}} = -\phi$ so that the resulting PSF has diffraction limited form (1.3). In practice, however, an accurate approximation of ϕ suffices.

In this paper, our focus is on the problem of estimating the phase from measurements of the wavefront gradient. We assume that the gradient data $\mathbf{g}$ is collected by a Shack–Hartmann sensor. The corresponding discrete phase $\boldsymbol{\phi}$ will then satisfy the stochastic linear equation

$$\mathbf{g} = \boldsymbol{\Gamma}\boldsymbol{\phi} + \mathbf{n}, \tag{1.6}$$

where $\boldsymbol{\Gamma}$ is a discrete gradient matrix, whose form is determined by the configuration of the Shack–Hartmann sensor, and $\mathbf{n}$ is the noise vector. Early and existing approaches to solving this problem involve minimizing the least squares function $||\boldsymbol{\Gamma}\boldsymbol{\phi} - \mathbf{g}||^2$. However, for large-scale adaptive optics systems, least squares solutions can yield unsatisfactory results, and the minimum variance estimator is preferred [6]. Minimum variance estimation is a Bayesian statistical approach in which a prior probability density is assumed on the phase. In our case, it can be accurately assumed that $\boldsymbol{\phi}$ is a realization of a Gaussian random vector with mean $\mathbf{0}$ and known covariance matrix $\boldsymbol{C_\phi}$. This, together with (1.6) and the assumption that the noise vector $\mathbf{n}$ is Gaussian with mean $\mathbf{0}$ and covariance matrix $\sigma^2\mathbf{I}$, yields a linear system of the form

$$(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma} + \sigma^2 \boldsymbol{C_\phi}^{-1})\boldsymbol{\phi} = \boldsymbol{\Gamma}^T\mathbf{g}. \tag{1.7}$$

The problem of efficiently solving (1.7), or, equivalently, of minimizing the penalized least squares function $||\boldsymbol{\Gamma}\boldsymbol{\phi}-\mathbf{g}||^2+\sigma^2\boldsymbol{\phi}^T\boldsymbol{C_\phi}^{-1}\boldsymbol{\phi}$, has seen much recent attention. An efficient direct method for the solution of (1.7) using sparse matrix techniques is explored in [6]. However, the most computationally efficient approaches have involved the use of multigrid to precondition conjugate gradient iterations [9, 10]. In this paper, we introduce two new approaches for approximately solving (1.7). The first involves the use of a symmetric positive definite approximation of $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}$ as a preconditioner for conjugate gradient iterations. The second approach is completely different and involves first computing the least squares solution of minimum norm $\boldsymbol{\phi}_{\mathrm{MNLS}} = \boldsymbol{\Gamma}^\dagger\mathbf{g}$, where "†" denotes pseudoinverse. The minimum norm solution is then denoised and stabilized via the solution of a linear system motivated by (1.7).

The paper is organized as follows. In section 2, we present the linear system that arises from the use of the Shack–Hartmann sensor; we derive the minimum variance linear system (1.7); and we discuss approximations of the covariance matrix $\boldsymbol{C_\phi}$. Computational methods are presented in section 3 and tested in section 4. We end with conclusions in section 5.

2. Wavefront reconstruction from discrete gradient measurements. In this section, we present the wavefront reconstruction problem that arises when the Shack–Hartmann wavefront sensor is used. The Shack–Hartmann sensor collects measurements of the gradient of incoming wavefronts of light emitted by the object being viewed by the telescope. It consists of an array of lenslets, each of which focuses the light within its aperture, and, typically, a charge coupled device (CCD) camera that records the position of the focal point of the light within each lenslet. A measurement of the average gradient of the wavefront over the lenslet aperture is then given by the position of the focal point. A schematic of the Shack–Hartmann sensor in one dimension is given in Figure 2.1. A more detailed description with further references can be found in [2].

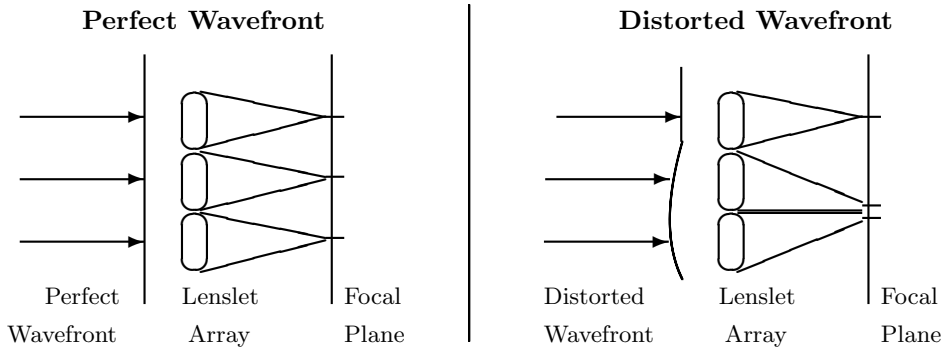


FIG. 2.1. *One-dimensional Shack–Hartman wavefront sensor schematic. The position of the focal points determines the average derivative of the wavefront, and hence of the phase, over each lenslet aperture.*

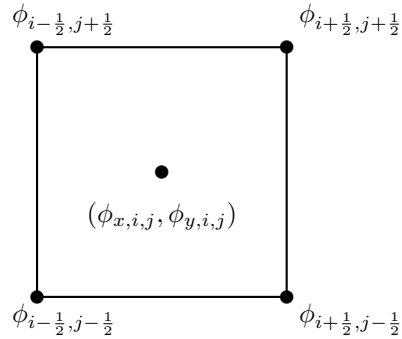


FIG. 2.2. *Fried geometry for the (i,j)th lenslet. The wavefront sensor yields a measurement of the gradient of the phase at (x_i, y_j). The values of the phase at the half grid points, i.e., at the corners of the apertures, are then sought.*

The standard computational methodology for reconstructing the phase from Shack–Hartmann gradient data was introduced by Fried in [4], where, from the gradient measurements, which are assumed to be centered within each lenslet array, the value of the phase at the corners is computed. This is the so-called Fried geometry and is illustrated in Figure 2.2. Assuming a square geometry and uniform computational grid with grid points $\{(x_i, y_j)\}_{i,j=1}^n$, and denoting $\nabla\phi(x_i, y_j) = (\phi_{x,i,j}, \phi_{y,i,j})$, we can approximate ϕ at the half grid points via the following two formulas:

$$\phi_{x,i,j} \approx \frac{1}{2}\left[(\phi_{i+\frac{1}{2},j-\frac{1}{2}} - \phi_{i-\frac{1}{2},j-\frac{1}{2}}) + (\phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i-\frac{1}{2},j+\frac{1}{2}})\right], \tag{2.1}$$

$$\phi_{y,i,j} \approx \frac{1}{2}\left[(\phi_{i-\frac{1}{2},j+\frac{1}{2}} - \phi_{i-\frac{1}{2},j-\frac{1}{2}}) + (\phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i+\frac{1}{2},j-\frac{1}{2}})\right]. \tag{2.2}$$

Here we have assumed a grid spacing $\Delta x = \Delta y = 1$. We note that in practice, the corners of the lenslet aperture frequently correspond to the points on the deformable mirror of the adaptive optics system where the deformations are actuated.

By column stacking the $n \times n$ array of the values of ϕ at the half grid points, one obtains an $n^2 \times 1$ vector $\boldsymbol{\phi}$. The equations on the right-hand side of (2.1) and (2.2) can then be written in matrix-vector forms $\boldsymbol{\Gamma}_x\boldsymbol{\phi}$ and $\boldsymbol{\Gamma}_y\boldsymbol{\phi}$, respectively. If the $n \times n \times 2$ array of gradient measurements is also column stacked, an $n^2 \times 2$ array

$$\begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

FIG. 2.3. *Grid representations of the Fried Laplacian (on the left) and the Hudgin Laplacian (on the right).*

$\mathbf{g}$ results. The unknown phase can then be reconstructed by solving the stochastic linear system

$$\mathbf{g} = \mathbf{\Gamma}\boldsymbol{\phi} + \mathbf{n}, \tag{2.3}$$

where $\mathbf{\Gamma} = [\mathbf{\Gamma}_x, \mathbf{\Gamma}_y]^T$ and $\mathbf{n} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$. Throughout the paper, the notation $\mathbf{y} \sim N(\boldsymbol{\mu}, \mathbf{C})$ will mean that $\mathbf{y}$ is a random draw from a Gaussian random vector with mean $\boldsymbol{\mu} \in \mathbb{R}^{n^2}$ and covariance matrix $\mathbf{C} \in \mathbb{R}^{n^2 \times n^2}$.

2.1. Least squares wavefront reconstruction. For small-scale wavefront reconstruction problems, the least squares solution of (2.3), given by minimizing

$$\min_{\phi} \|\mathbf{\Gamma}\boldsymbol{\phi} - \mathbf{g}\|_2^2,$$

is known to provide excellent results [6]. One can equivalently solve the normal equations

$$\mathbf{\Gamma}^T\mathbf{\Gamma}\boldsymbol{\phi} = \mathbf{\Gamma}^T\mathbf{g}. \tag{2.4}$$

The matrix $\mathbf{\Gamma}^T\mathbf{\Gamma}$ corresponds to a nonstandard discretization of the Laplacian operator Δ, with the grid representation given on the left in Figure 2.3 and homogeneous Neumann boundary conditions. We will call this the Fried discrete Laplacian. The standard discretization of the Laplacian has the grid representation given on the right in Figure 2.3. We note that this is the discrete Laplacian that results from what is known in the adaptive optics community as Hudgin geometry [13] and is therefore referred to as the Hudgin discrete Laplacian [19]. We will use this terminology in what follows.

Fried geometry yields more robust phase estimates than does Hudgin geometry [20]. However, the null-space of the Fried Laplacian is larger than that of the Hudgin Laplacian. In particular, it contains what is known as the waffle mode, which is the $n \times n$ array with entries

$$[\boldsymbol{\phi}_{\mathrm{WM}}]_{ij} = (-1)^{i+j}. \tag{2.5}$$

In what follows, we will also use $\boldsymbol{\phi}_{\mathrm{WM}}$ to denote the corresponding $n^2 \times 1$ column stacked vector. We say that a vector ϕ contains waffle mode if

$$\boldsymbol{\phi}^T\boldsymbol{\phi}_{\mathrm{WM}} \neq 0.$$

Waffle mode has been observed in operational adaptive optics systems [14], and hence, its presence in the null-space of the Fried Laplacian is not of only academic interest.

Before continuing, we prove an interesting relationship between $\boldsymbol{\phi}_{\mathrm{WM}}$ and the Hudgin discrete Laplacian with homogeneous Dirichlet, homogeneous Neumann, and periodic boundary conditions.

THEOREM 2.1. $n^2 \times n^2$ $\mathbf{L}$, $\boldsymbol{\phi}_{\rm WM}$

$$\max_{\boldsymbol{\phi}} \frac{\boldsymbol{\phi}^T \mathbf{L} \boldsymbol{\phi}}{\|\boldsymbol{\phi}\|^2}. \tag{2.6}$$

$\boldsymbol{\phi}_{\rm WM}$ (2.6) $n \to \infty$

. First, by Gerschgorin's circle theorem, the eigenvalues of $\mathbf{L}$ satisfy $0 \leq \lambda \leq 8$ for each of the three types of boundary conditions. Thus $0 \leq \boldsymbol{\phi}^T \mathbf{L} \boldsymbol{\phi} / \|\boldsymbol{\phi}\|^2 \leq 8$.

In the case of both periodic and homogeneous Neumann boundary conditions, $\boldsymbol{\phi}_{\rm WM}^T \mathbf{L} \boldsymbol{\phi}_{\rm WM} / \|\boldsymbol{\phi}_{\rm WM}\|^2 = 8$, and hence $\boldsymbol{\phi}_{\rm WM}$ solves (2.6).

In the case of homogeneous Dirichlet boundary conditions, we note that $\mathbf{L}$ differs from the discrete Laplacian with periodic boundary conditions in that it has 0 in place of -1 in $4n$ locations. A straightforward calculation together with the fact that $\|\boldsymbol{\phi}_{\rm WM}\|^2 = n^2$ then yields $\boldsymbol{\phi}_{\rm WM}^T \mathbf{L} \boldsymbol{\phi}_{\rm WM} / \|\boldsymbol{\phi}_{\rm WM}\|^2 = 8 - 4/n \to 8$, and hence $\boldsymbol{\phi}_{\rm WM}$ converges to the solution of (2.6) as $n \to \infty$. □

We note that for large-scale problems $8 - 4/n \approx 8$, and hence, $\boldsymbol{\phi}_{\rm WM}$ maximizes, or nearly maximizes, (2.6) in all three cases. This suggests the use of regularization by the Laplacian to remove waffle mode and other high frequency errors in the phase estimates. As we will see in the next section, such an approach can be motivated statistically when the minimum variance approach is taken.

2.2. Minimum variance wavefront reconstruction. The preferred approach for stabilizing least squares phase estimation is to compute a minimum variance estimate for $\boldsymbol{\phi}$ (c.f. [18]). Minimum variance estimation can be viewed as the analogue of least squares estimation in the Bayesian setting. As we will see, the resulting equations are similar. In the minimum variance framework, we assume that the phase satisfies

$$\boldsymbol{\phi} \sim N(\mathbf{0}, \boldsymbol{C}_{\boldsymbol{\phi}}), \tag{2.7}$$

where the covariance $\boldsymbol{C}_{\boldsymbol{\phi}}$ is specified a priori. The minimum variance estimator can then be defined as follows.

DEFINITION 2.2. $\boldsymbol{\phi}$ $\mathbf{g}$

$$\boldsymbol{\phi}_{\rm MV} = \hat{\mathbf{B}} \mathbf{g},$$

$$\hat{\mathbf{B}} = \arg \min_{\mathbf{B} \in \mathbb{R}^{n \times m}} E(\|\mathbf{B}\mathbf{g} - \boldsymbol{\phi}\|^2).$$

E

In our case, the minimum variance estimator has an elegant closed form, which we state in the next theorem. Standard proofs of this theorem use notation from probability theory. Here we present a proof, outlined in [18], from a matrix analysis viewpoint.

THEOREM 2.3. $\mathbf{g}$ (2.3) $\mathbf{n} \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$ $\boldsymbol{\phi} \sim N(\mathbf{0}, \boldsymbol{C}_{\boldsymbol{\phi}})$ $\boldsymbol{C}_{\boldsymbol{\phi}}$ $\mathbf{n}$ $\boldsymbol{\phi}$ $\boldsymbol{\phi}$ $\mathbf{g}$

$$\boldsymbol{\phi}_{\rm MV}^{\sigma} = \left(\boldsymbol{\Gamma}^T \boldsymbol{\Gamma} + \sigma^2 \boldsymbol{C}_{\boldsymbol{\phi}}^{-1} \right)^{-1} \boldsymbol{\Gamma}^T \mathbf{g}. \tag{2.8}$$

. First, we note that

$$\begin{aligned} E(\|\mathbf{Bg}-\boldsymbol{\phi}\|^2) &= \text{trace}\left(E[(\mathbf{Bg}-\boldsymbol{\phi})(\mathbf{Bg}-\boldsymbol{\phi})^T]\right) \\ &= \text{trace}\left(\mathbf{B}E[\mathbf{gg}^T]\mathbf{B}^T - \mathbf{B}E[\mathbf{g}\boldsymbol{\phi}^T] - E[\boldsymbol{\phi}\mathbf{g}^T]\mathbf{B}^T + E[\boldsymbol{\phi}\boldsymbol{\phi}^T]\right). \end{aligned}$$

Then, using the distributive property of the trace function and the identity

$$\frac{d}{d\mathbf{B}}\text{trace}(\mathbf{B}^T\mathbf{C}) = \left(\frac{d}{d\mathbf{B}}\text{trace}(\mathbf{BC})\right)^T = \mathbf{C},$$

we see that $dE(\|\mathbf{Bg}-\boldsymbol{\phi}\|^2)/d\mathbf{B} = \mathbf{0}$ when

$$\hat{\mathbf{B}} = E[\boldsymbol{\phi}\mathbf{g}^T]E[\mathbf{gg}^T]^{-1}.$$

Now, since $\boldsymbol{\phi}$ and $\mathbf{n}$ are independent, $E[\boldsymbol{\phi}\mathbf{n}^T] = E[\mathbf{n}\boldsymbol{\phi}^T] = \mathbf{0}$. Hence, using (2.3), we obtain

$$\begin{aligned} E[\boldsymbol{\phi}\mathbf{g}^T] &= E[\boldsymbol{\phi}(\boldsymbol{\Gamma}\boldsymbol{\phi}+\mathbf{n})^T] \\ &= E[\boldsymbol{\phi}\boldsymbol{\phi}^T]\boldsymbol{\Gamma}^T. \end{aligned}$$

Similarly,

$$\begin{aligned} E[\mathbf{gg}^T] &= E[(\boldsymbol{\Gamma}\boldsymbol{\phi}+\mathbf{n})(\boldsymbol{\Gamma}\boldsymbol{\phi}+\mathbf{n})^T] \\ &= \boldsymbol{\Gamma}E[\boldsymbol{\phi}\boldsymbol{\phi}^T]\boldsymbol{\Gamma}^T + E[\mathbf{nn}^T]. \end{aligned}$$

Thus, since $E[\boldsymbol{\phi}\boldsymbol{\phi}^T] = \boldsymbol{C}_{\boldsymbol{\phi}}$ and $E[\mathbf{nn}^T] = \sigma^2\mathbf{I}$, we have

$$\begin{aligned} \hat{\mathbf{B}}\mathbf{g} &= \boldsymbol{C}_{\boldsymbol{\phi}}\boldsymbol{\Gamma}^T\left(\boldsymbol{\Gamma}^T\boldsymbol{C}_{\boldsymbol{\phi}}\boldsymbol{\Gamma}+\sigma^2\mathbf{I}\right)^{-1}\mathbf{g} \\ &= \left(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}+\sigma^2{\boldsymbol{C}_{\boldsymbol{\phi}}}^{-1}\right)^{-1}\boldsymbol{\Gamma}^T\mathbf{g}. \end{aligned}$$

The last equality follows from straightforward algebraic manipulation. □

Thus the minimum variance wavefront estimate can be obtained by solving the linear system

$$\left(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}+\sigma^2{\boldsymbol{C}_{\boldsymbol{\phi}}}^{-1}\right)\boldsymbol{\phi} = \boldsymbol{\Gamma}^T\mathbf{g} \tag{2.9}$$

or, equivalently, by minimizing the penalized least squares function

$$\|\boldsymbol{\Gamma}\boldsymbol{\phi}-\mathbf{g}\|_2^2 + \sigma^2\boldsymbol{\phi}^T{\boldsymbol{C}_{\boldsymbol{\phi}}}^{-1}\boldsymbol{\phi}. \tag{2.10}$$

Note, then, that the minimum variance estimator can be viewed as a Tikhonov estimator, with quadratic regularization term $\sigma^2\boldsymbol{\phi}^T{\boldsymbol{C}_{\boldsymbol{\phi}}}^{-1}\boldsymbol{\phi}$.

2.3. Incorporating the telescope's pupil. For simplicity of implementation, computations are typically done on a square computational grid, even though the telescope pupil geometry is usually either circular or annular. This information is incorporated into the problem formulation using the pupil mask matrix $\mathbf{M}$ defined by

$$[\mathbf{M}]_{ii} = \begin{cases} 1, & i \text{ inside the pupil,} \\ 0 & \text{otherwise.} \end{cases}$$

We then modify the linear stochastic model (2.3) as follows:

$$\mathbf{Mg} = \mathbf{M\Gamma}\boldsymbol{\phi} + \mathbf{n}, \tag{2.11}$$

where $\mathbf{M\Gamma} \stackrel{\text{def}}{=} [\mathbf{M\Gamma}_x, \mathbf{M\Gamma}_y]^T$. Following the minimum variance approach outlined above, the penalized least squares function (2.10) takes the form

$$\|\mathbf{M}(\mathbf{\Gamma}\boldsymbol{\phi} - \mathbf{g})\|_2^2 + \sigma^2 \boldsymbol{\phi}^T {\boldsymbol{C}_{\boldsymbol{\phi}}}^{-1} \boldsymbol{\phi}, \tag{2.12}$$

and the linear system (2.9) is re-expressed as

$$\left(\mathbf{\Gamma}^T \mathbf{M\Gamma} + \sigma^2 {\boldsymbol{C}_{\boldsymbol{\phi}}}^{-1}\right) \boldsymbol{\phi} = \mathbf{\Gamma}^T \mathbf{Mg}. \tag{2.13}$$

The corresponding regular least squares normal equations are given by

$$\mathbf{\Gamma}^T \mathbf{M\Gamma}\boldsymbol{\phi} = \mathbf{\Gamma}^T \mathbf{Mg}. \tag{2.14}$$

2.4. Approximating the phase covariance. The phase covariance $\boldsymbol{C}_{\boldsymbol{\phi}}$ must be chosen so that it is both physically realistic and amenable to fast computational methods. We call attention to the fact that for an actual adaptive optics system, algorithms for solving (2.9) must do so in real time.

Perhaps the most standard approximation for $\boldsymbol{C}_{\boldsymbol{\phi}}$ is to assume that it has the form

$$\boldsymbol{C}_{\boldsymbol{\phi}}^{\mathbf{VK}} = \mathbf{F}^* \mathbf{\Lambda} \mathbf{F}, \tag{2.15}$$

where $\mathbf{F}$ is the two-dimensional discrete Fourier transform matrix, "$*$" denotes conjugate transpose, and the matrix $\mathbf{\Lambda}$ is diagonal with entries coming from the von Karman spatial power spectral density of the atmospheric refractive index fluctuations, with universal $-11/3$ inverse power law:

$$[\mathbf{\Lambda}]_{k,k} = \frac{c^2}{[|k|^2 + 1/L_0^2]^{11/6}}. \tag{2.16}$$

Here k denotes spatial frequency, L_0 is the turbulence outer-scale, which prevents an unphysically infinite amount of energy at the origin, and c is the phase screen strength (c.f. [16]).

However, given the desire for real time computations for large-scale lenslet arrays on very high-order adaptive optics systems, a sparse covariance approximation is desirable. Such an approximation was introduced in [6]. It exploits the fact that

$$|k|^{-11/3} \approx |k|^{-4}.$$

In particular, using the fact that the biharmonic, or squared Laplacian, operator "Δ^2" has spectrum $|k|^4$, the following discrete approximation of the covariance can be used:

$$\boldsymbol{C}_{\boldsymbol{\phi}}^{\mathbf{BH}} = c_0 \, \mathbf{L}^{-2}, \tag{2.17}$$

where $\mathbf{L}$ is a discrete Laplacian matrix. The constant c_0 in (2.17) can be physically interpreted as the strength of the turbulence and is chosen in our simulations so that

$$E[\boldsymbol{\phi}^T (\boldsymbol{C}_{\boldsymbol{\phi}}^{\mathbf{VK}})^{-1} \boldsymbol{\phi}] = E[\boldsymbol{\phi}^T (\boldsymbol{C}_{\boldsymbol{\phi}}^{\mathbf{BH}})^{-1} \boldsymbol{\phi}] \tag{2.18}$$

holds.

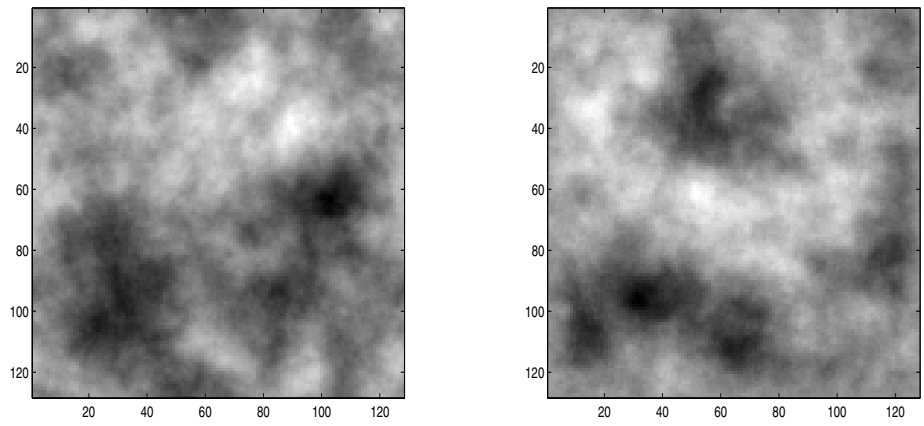


FIG. 2.4. *Random draws from zero mean Gaussian random vectors with (on the left) von Karman covariance defined by* (2.15) *and (on the right) inverse of squared-Laplacian covariance defined in* (2.17).

In order to visually compare the two covariance approximations (2.15) and (2.17), we plot random draws from the zero mean Gaussian random vectors with these as the covariance matrices in Figure 2.4. They clearly exhibit similar characteristics.

REMARK 1. (2.17) (2.12)
ℓ^2
$(\sigma^2/c_0)\|\mathbf{L}\boldsymbol{\phi}\|^2$ $\|\mathbf{M}(\boldsymbol{\Gamma}\boldsymbol{\phi}-\mathbf{g})\|^2$
2.1

3. Numerical methods. In the early papers of Fried [4] and Hudgin [13], the Gauss–Seidel iteration was used for numerically solving the normal equations (2.14). In [17], symmetric Gauss–Seidel is implemented. However, it is well known that iterative methods such these converge slowly in practice. Such methods are effective, however, if used within a multigrid framework or as preconditioners (given that they are symmetric) for conjugate gradient iterations. The use of multigrid for solving (2.14) is explored in [1, 15].

Direct methods for (2.14) are made feasible by the fact that $\boldsymbol{\Gamma}^T\mathbf{M}\boldsymbol{\Gamma}$ is very sparse and is fixed for a specific telescope. In [12], the least squares solution of minimum norm is computed via the pseudoinverse of $\mathbf{M}\boldsymbol{\Gamma}$, which we denote $(\mathbf{M}\boldsymbol{\Gamma})^\dagger$. The pseudoinverse can be efficiently approximated using the Cholesky factorization. In particular, noting that

$$(\mathbf{M}\boldsymbol{\Gamma})^\dagger = \lim_{\epsilon\to 0^+} (\boldsymbol{\Gamma}^T\mathbf{M}\boldsymbol{\Gamma} + \epsilon\,\mathbf{I})^{-1}\boldsymbol{\Gamma}^T\mathbf{M},$$

one can compute a Cholesky factorization of $\boldsymbol{\Gamma}^T\mathbf{M}\boldsymbol{\Gamma} + \epsilon\,\mathbf{I}$ for ϵ small, e.g., the square root of machine epsilon ($\approx 10^{-8}$). Because it will be useful to us later, we give a detailed description of this approach now. After performing a reordering of indices using MATLAB's `symamd` function, $\boldsymbol{\Gamma}^T\mathbf{M}\boldsymbol{\Gamma} + \epsilon\,\mathbf{I}$ has the form

$$\tilde{\mathbf{A}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{A} + \epsilon\,\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \epsilon\,\mathbf{I} \end{bmatrix}, \tag{3.1}$$

where $\mathbf{A}$ is sparse and symmetric positive semidefinite. We can then compute the Cholesky factorization $\tilde{\mathbf{A}} = \mathbf{C}^T\mathbf{C}$. Assuming, without loss of generality, that $\tilde{\mathbf{A}} = \mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma}+\epsilon\mathbf{I}$, the minimum norm least squares solution can be efficiently and accurately approximated via

$$\boldsymbol{\phi}_{\text{MNLS}} = (\mathbf{M}\mathbf{\Gamma})^{\dagger}\mathbf{M}\mathbf{g} \approx \mathbf{C}^{-1}\mathbf{C}^{-T}\mathbf{\Gamma}^T\mathbf{M}\mathbf{g}. \tag{3.2}$$

Finally, since $\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma}$ depends only on the inherent structure of the telescope, the above Cholesky factorization can be computed offline, and hence, the cost of computing minimum norm least squares solutions using this approach is restricted to the computation of $\mathbf{\Gamma}^T\mathbf{M}\mathbf{g}$ and to the applications of $\mathbf{C}^{-1}$ and $\mathbf{C}^{-T}$. Also, due to the presence of the pupil mask matrix $\mathbf{M}$ and the use of a sparse reordering of indices, the Cholesky factorization is very sparse, resulting in very efficient computations of $\boldsymbol{\phi}_{\text{MNLS}}$.

In [6], it is shown that for an $N \times N$ Hudgin Laplacian, the scaling law for the number of floating point operations (flops) needed for its Cholesky factorization (after sparse reordering) is $N^{3/2}$. For the Fried Laplacian, the number of flops needed for its Cholesky factorization (again after sparse reordering) should be very nearly the same. Furthermore, the presence of the pupil will further reduce computational requirements. We emphasize again that this Cholesky factorization need only be computed once.

For adaptive optics systems with extremely large lenslet arrays, the minimum variance estimate $\boldsymbol{\phi}^{\sigma}_{\text{MV}}$ obtained by solving (2.13) is preferable to $\boldsymbol{\phi}_{\text{MNLS}}$. An added difficulty arises, however, due to the presence of $\boldsymbol{C}_{\boldsymbol{\phi}}$. Until recently, the von Karman covariance approximation (2.15), (2.16) was standard. The fact that this is a full matrix made direct solutions of (2.13) infeasible. However, the sparse biharmonic approximation (2.17) presented in [6] allowed for a direct approach using a Cholesky factorization of

$$\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma} + (\sigma^2/c_0)\mathbf{L}^2, \tag{3.3}$$

where $\mathbf{L}$ is a discretized Laplacian matrix. However, the Cholesky factorization must be recomputed as σ^2/c_0 changes, which makes the direct approach less desirable for minimum variance estimation.

3.1. Preconditioned conjugate gradient methods. The preconditioned conjugate gradient method (PCG) is an iterative method for minimizing quadratic functions with symmetric positive semidefinite Hessian matrices [18], such as is the case for (2.12). The implementation of PCG requires the solution of a linear system of the form

$$\mathbf{P}\mathbf{z} = \mathbf{v} \tag{3.4}$$

at each iteration, where $\mathbf{P}$ is the symmetric positive definite preconditioning matrix or, simply, the preconditioner. For the resulting implementation of PCG to be efficient, solutions of (3.4) must be efficiently computable.

Thus far, the most computationally efficient approach for minimizing (2.12) is to let $\mathbf{z}$ in (3.4) be what results following the application of one multigrid v-cycle [3] applied to the linear system

$$\left(\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma} + \sigma^2{\boldsymbol{C}_{\boldsymbol{\phi}}}^{-1}\right)\mathbf{z} = \mathbf{v}.$$

The corresponding preconditioning matrix $\mathbf{P}$ is made symmetric either by using a symmetric smoother such as Jacobi or symmetric Gauss–Seidel or by using Gauss–Seidel with forward substitution for the presmoothing iterations and an equal number of Gauss–Seidel iterations with backward substitution for the postsmoothing step. The resulting algorithm, which we denote MGPCG (multigrid PCG), was applied for the von Karman covariance approximation (2.15), (2.16) in [10] and for the biharmonic covariance approximation (2.17) in [9], with further analysis in [19]. The sparsity of the discrete biharmonic makes the latter implementation the more efficient of the two. The effectiveness of multigrid in this setting is not surprising when one considers that $\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma}$ is a discrete Laplacian matrix and that $\boldsymbol{C_\phi}^{-1}$ either is or is well-approximated by a discrete biharmonic matrix. Multigrid is known to be very effective for solving linear systems involving both the discrete Laplacian and the discrete biharmonic matrices.

A preconditioner that has not been used for PCG applied to the problem of minimizing (2.12) is

$$\mathbf{P} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{A} + \epsilon\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{3.5}$$

where $\mathbf{A}$ is as defined in (3.1). As in that case, we compute a Cholesky factorization $\mathbf{C}^T\mathbf{C}$ of $\mathbf{P}$. Since $\mathbf{P}$ is noise independent, this can be done off-line. Thus the application of $\mathbf{P}^{-1}$ requires only one forward and one backward substitution. We will call $\mathbf{P}$ in (3.5) the least squares preconditioner, since in this case $\mathbf{P}$ is the coefficient matrix for the regular least squares normal equations, and the resulting method the least squares PCG (LSPCG).

3.2. The denoised least squares method. In this subsection, we introduce a new approach for obtaining approximate solutions of (2.13). First, we denote the Hudgin discrete Laplacian with periodic boundary conditions by $\mathbf{L}$. Then, we have

$$\mathbf{L} = \mathbf{F}^*\text{diag}(\lambda_1, \ldots, \lambda_{n^2})\mathbf{F}, \tag{3.6}$$

where $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n^2}$ are the eigenvalues of $\mathbf{L}$, and $\mathbf{F}$ denotes the discrete Fourier transform matrix with $\mathbf{F}^*$ its conjugate transpose. We note that then $\lambda_1 = 0$ corresponds to the constant eigenvector of $\mathbf{L}$, and $\lambda_{n^2} = 8$ corresponds to the waffle mode eigenvector $\boldsymbol{\phi}_{\text{WM}}$ (recall Theorem 2.1).

Multiplying both sides of (2.13) by the pseudoinverse of $\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma}$, we obtain

$$((\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma})^\dagger\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma} + (\sigma^2/c_0)(\mathbf{\Gamma}^T\mathbf{M}\mathbf{\Gamma})^\dagger\mathbf{L}^2)\boldsymbol{\phi} = \boldsymbol{\phi}_{\text{MNLS}}, \tag{3.7}$$

which can in turn be solved to obtain a smoothed, or denoised, approximation of $\boldsymbol{\phi}_{\text{MNLS}}$.

We ignore for the moment the pupil mask matrix $\mathbf{M}$ and consider the matrix product $(\mathbf{\Gamma}^T\mathbf{\Gamma})^\dagger\mathbf{L}$. Recall that the Fried discrete Laplacian $\mathbf{\Gamma}^T\mathbf{\Gamma}$ has the grid representation given in Figure 2.2. Note that after a rotation of the computational grid by $\pi/4$ radians, the grid representation of $\mathbf{\Gamma}^T\mathbf{\Gamma}$ will match that of a Hudgin discrete Laplacian, but with a grid spacing that is larger by a factor of $\sqrt{2}$. Taking this into account, we approximate the Fried discrete Laplacian as follows:

$$\mathbf{\Gamma}^T\mathbf{\Gamma} \approx \mathbf{F}^*\text{diag}(\lambda_1, \ldots, \lambda_r, 0, \ldots, 0)\mathbf{F}, \tag{3.8}$$

where the λ_i's are as in (3.6), and r chosen so that

$$\lambda_r < 8 - \frac{1}{4\sqrt{2}} \leq \lambda_{r+1}. \tag{3.9}$$

Note that $\boldsymbol{\phi}_{\mathrm{WM}}$ has a period of 4 on the computational grid and of $4\sqrt{2}$ on the rotated grid—hence our choice of truncation rule. Furthermore, our own computational experiments indicate that (3.9) is optimal in the sense that it minimizes the error in the phase reconstructions that result when the method we now present is used.

From (3.8), we know that within the telescope's pupil,

$$\boldsymbol{\Gamma}^T\mathbf{M}\boldsymbol{\Gamma} \approx \mathbf{F}^*\mathrm{diag}(\lambda_1,\ldots,\lambda_r,0,\ldots,0)\mathbf{F}. \tag{3.10}$$

Hence,

$$(\boldsymbol{\Gamma}^T\mathbf{M}\boldsymbol{\Gamma})^\dagger\boldsymbol{\Gamma}^T\mathbf{M}\boldsymbol{\Gamma} \approx \mathbf{F}^*\,\mathrm{diag}(1_1,\ldots,1_r,0,\ldots,0)\mathbf{F}, \tag{3.11}$$

$$(\boldsymbol{\Gamma}^T\mathbf{M}\boldsymbol{\Gamma})^\dagger\mathbf{L}^2 \approx \mathbf{F}^*\mathrm{diag}(\lambda_1,\ldots,\lambda_r,0,\ldots,0)\mathbf{F}. \tag{3.12}$$

This leads to the following approximation of (3.7):

$$\mathbf{F}^*\mathbf{D}\mathbf{F}\boldsymbol{\phi} = \boldsymbol{\phi}_{\mathrm{MNLS}}, \tag{3.13}$$

where $\mathbf{D}$ is diagonal with elements

$$[\mathbf{D}]_{ii} = \begin{cases} 1+(\sigma^2/c_0)\lambda_i, & \lambda_i < r, \\ 0, & \lambda_i \geq r. \end{cases} \tag{3.14}$$

Note that (3.13), (3.14) bears some resemblance to a filtered SVD approximation [18, 11] of (3.7).

The denoised minimum norm least squares solution can then be efficiently computed via

$$\boldsymbol{\phi}_{\mathrm{DMNLS}} = \mathbf{F}^*\mathbf{D}^\dagger\mathbf{F}\boldsymbol{\phi}_{\mathrm{MNLS}}, \tag{3.15}$$

where $\mathbf{D}^\dagger$ is the pseudoinverse of $\mathbf{D}$.

As we will see, this approach is effective in practice. However, it also has the benefit of being simple to implement and very computationally efficient. Furthermore, it can be easily incorporated into adaptive optics systems that compute regular least squares solutions other than $\boldsymbol{\phi}_{\mathrm{MNLS}}$. In particular, a general least squares solution $\boldsymbol{\phi}_{\mathrm{LS}}$ can be denoised via

$$\boldsymbol{\phi}_{\mathrm{GDLS}} = \mathbf{F}^*\mathbf{D}^\dagger\mathbf{F}\boldsymbol{\phi}_{\mathrm{LS}}, \tag{3.16}$$

where "GDLS" stands for "gradient denoised least squares." In the next theorem, we show that $\boldsymbol{\phi}_{\mathrm{GDLS}}$ will not contain waffle mode even if $\boldsymbol{\phi}_{\mathrm{LS}}$ does. This suggests that (3.16) should be considered as a method for removing waffle mode from least squares solutions in operational adaptive optics systems [14].

We end the section with a proof that the least squares solution of minimum norm, the minimum variance solution, and the denoised least squares solution do not contain waffle mode.

THEOREM 3.1. [illegible] $\phi_{\mathrm{MNLS}} = (\mathbf{M}\boldsymbol{\Gamma})^\dagger\mathbf{M}\mathbf{g}$, [illegible] $\boldsymbol{\phi}^{\sigma}_{\mathrm{MV}}$ [illegible] (2.8), [illegible] [illegible] (3.16) [illegible] $\boldsymbol{\phi}_{\mathrm{LS}}$ [illegible]. The null-space of $\mathbf{M}\boldsymbol{\Gamma}$ contains $\boldsymbol{\phi}_{\mathrm{WM}}$. The result for $\boldsymbol{\phi}_{\mathrm{MNLS}}$ then follows from the fact that the range of $(\mathbf{M}\boldsymbol{\Gamma})^\dagger$ and the null-space $\mathbf{M}\boldsymbol{\Gamma}$ only trivially intersect.

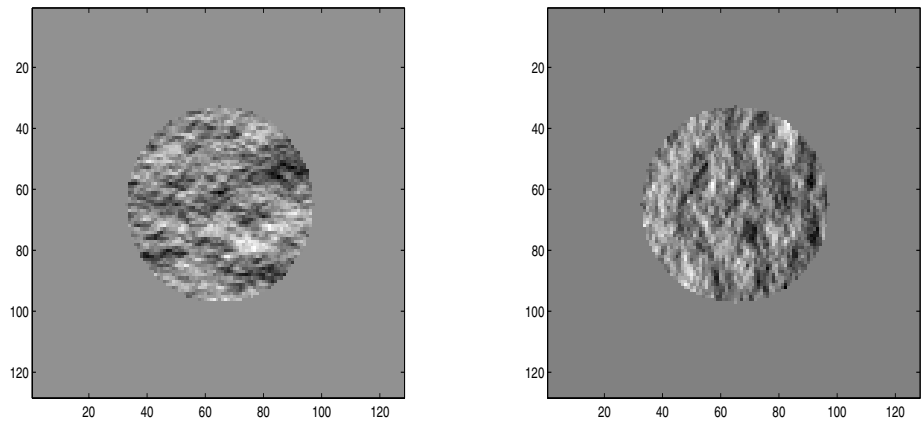


FIG. 4.1. *Noisy gradient data. The x and y components of the gradient data computed via* (2.3) *are seen on the left and right, respectively.*

For the minimum variance solution, we note that $\boldsymbol{\phi}_{\mathrm{WM}}^T \boldsymbol{C}_{\boldsymbol{\phi}}{}^{-1} \boldsymbol{\phi}_{\mathrm{WM}} > 0$ for either choice of $\boldsymbol{C}_{\boldsymbol{\phi}}$. Noting that $\mathbf{M\Gamma}\boldsymbol{\phi}_{\mathrm{WM}} = \mathbf{0}$, it follows immediately that the minimizer of (2.12) does not contain waffle mode.

From (3.16), (3.14), (3.9), and the fact that $\boldsymbol{\phi}_{\mathrm{WM}}$ is an eigenvector of $\mathbf{L}$ with eigenvalue 8, it follows that $\mathbf{F}^*\mathbf{D}^\dagger\mathbf{F}\boldsymbol{\phi}_{\mathrm{WM}} = \mathbf{0}$. Thus $\boldsymbol{\phi}_{\mathrm{DLS}}$ will not contain waffle mode. □

4. Numerical experiments. We now test the effectiveness of the above approaches on simulated Shack–Hartman sensor data. We simulate a phase profile by taking a random draw from the Gaussian random vector $N(\mathbf{0}, \boldsymbol{C}_{\boldsymbol{\phi}}^{\mathbf{KV}})$ with physically realistic values for the parameters in (2.16). The phase screen used in our first experiment is the 128×128 array plotted on the left in Figure 2.4. The noisy gradient data shown in Figure 4.1 was obtained using (2.11) with $\mathbf{n}$ an i.i.d. Gaussian random vector with variance chosen so that the signal-to-noise ratio (SNR) is 20, which is what was used in [19]. Note that

$$\mathrm{SNR} := \frac{\|\mathbf{g}\|^2}{E(\|\mathbf{n}\|^2)} = \frac{\|\mathbf{g}\|^2}{\mathrm{trace}(E[\mathbf{n}\mathbf{n}^T])} = \frac{\|\mathbf{g}\|^2}{2\,N_{\mathrm{pupil}}\,\sigma^2},$$

where N_{pupil} is the number of pixels within the telescope's pupil. For a comprehensive comparison, we also generate noisy gradient data at SNRs of 50, 10, and 5.

In our comparisons, we apply LSPCG and MGPCG to problem (2.12) with covariance (2.17). Our implementation of MGPCG used two iterations of Gauss–Seidel with forward substitution for the presmoothing iterations and two iterations of Gauss–Seidel with backward substitution for the postsmoothing iterations. The grid transfer operator used full-weighting with homogeneous Dirichlet boundary conditions for the restriction and a constant times its transpose for interpolation. See [3] for details on this implementation. We also apply the GDLS method. In all cases, c_0 was computed using (2.18).

Reconstructions at SNR = 20 are given in Figure 4.2, and convergence results are plotted in Figure 4.3. We note that more than two iterations of MGPCG and more

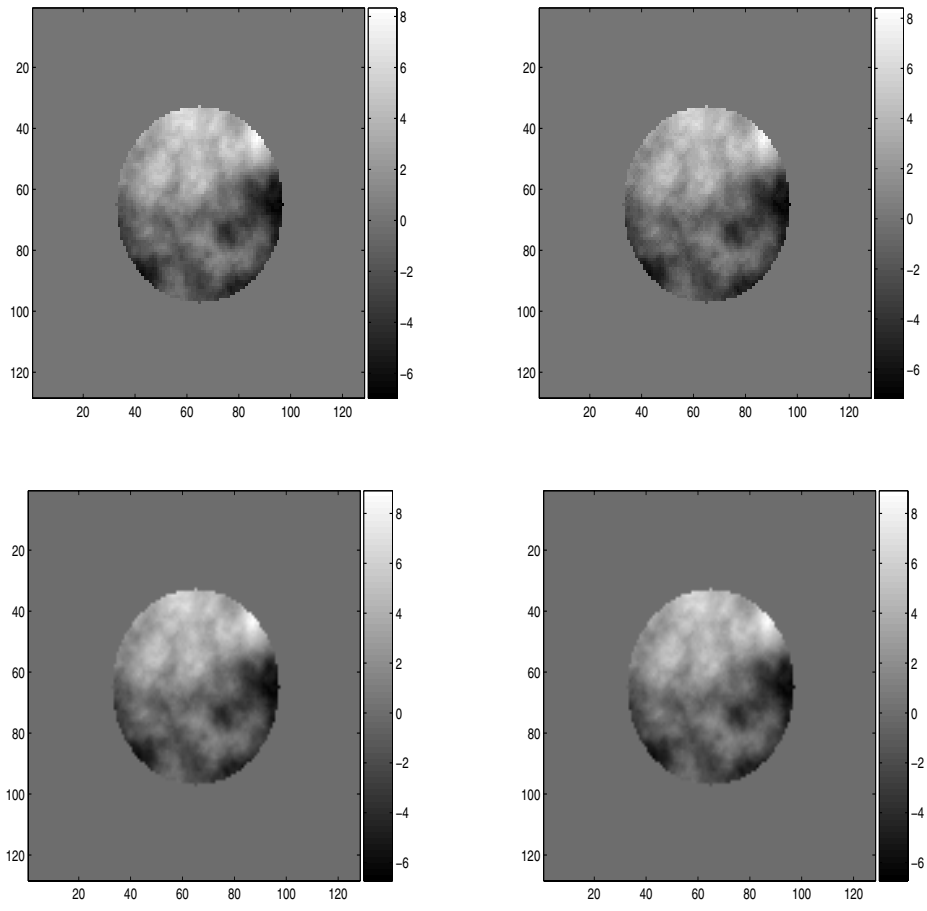


FIG. 4.2. *True phase on top*; *reconstruction obtained with one iteration of LSPCG on the upper right*; *reconstruction obtained with three iterations of MGPCG iterations on the lower left*; *and reconstruction obtained using GDLS on the lower right.*

than one iteration of LSPCG result in, effectively, no additional reduction in relative error. In each plot, we denote the GDLS curve as constant due to the fact that it yields a single estimate. Since it is accuracy in the approximation of the true phase $\boldsymbol{\phi}_{\text{true}}$ that we are concerned with, we plot the relative error

$$\frac{||\boldsymbol{\phi}_* - \boldsymbol{\phi}_{\text{true}}||}{||\boldsymbol{\phi}_{\text{true}}||},$$

where $\boldsymbol{\phi}_*$ is the phase estimate. For the PCG methods, $\boldsymbol{\phi}_* = \boldsymbol{\phi}_k$, where k is the PCG iteration index, whereas for GDLS, $\boldsymbol{\phi}_* = \boldsymbol{\phi}_{\text{DMNLS}}$.

The results in Figure 4.3 indicate that while two iterations of MGPCG yields the smallest relative error in every case, both LSPCG and GDLS are competitive. Furthermore, the reconstructions obtained by GDLS and by one iteration of LSPCG are very similar to those obtained by MGPCG. This can be seen, for SNR = 20, in Figure 4.2, where the true phase and the reconstructions obtained using a single

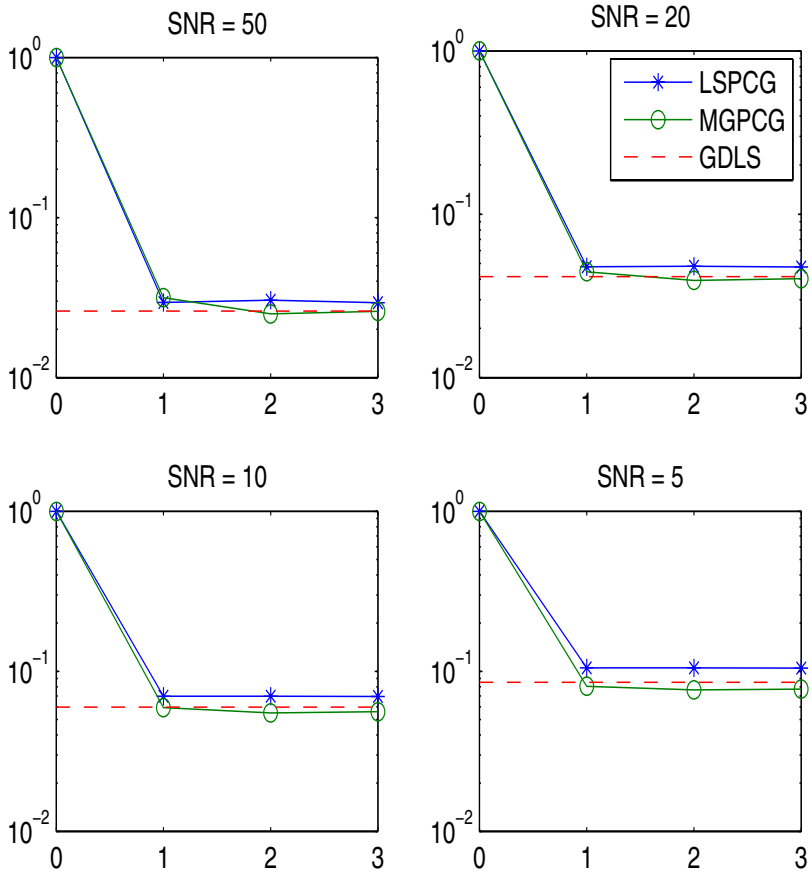


FIG. 4.3. *Relative error plots for LSPCG, MGPCG, and GDLS at SNRs of* 50, 20, 10, *and* 5.

iteration of LSPCG and three iterations of MGPCG are given.

In Table 4.1 we give the average CPU time for each method over three consecutive runs on $n \times n$ grids with $n = 32, 64, 128, 256$, and 512. In each case the SNR was taken to be 20. This table shows that both the LSPCG and GDLS solutions can be obtained much more efficiently than the MGPCG solutions—at least using our implementations of these methods—and that the computation of $\boldsymbol{\phi}_{\text{DMNLS}}$ is the most efficient. In fact, the values in the table indicate that the GDLS estimates are obtained 50–100 times more efficiently than those obtained using two iterations of MGPCG. This together with the facts that GDLS produces reconstructions with low relative error (see Figure 4.3) and that GDLS and MGPCG reconstructions are very similar (see Figure 4.2) suggest that GDLS is an approach that deserves attention.

We note that the choice of the discrete Fourier transform in the decomposition (3.10) will yield ringing artifacts at the boundaries in the phase reconstructions. We do not see this in the reconstructions presented here because there is sufficient distance between the pupil boundary and the boundary of the computational domain (see

TABLE 4.1
CPU times in seconds for various methods on an $n \times n$ computational grid.

n	MGPCG 1	MGPCG 2	LSPCG	GDLS
32	0.16	0.20	0.023	0.0020
64	0.27	0.38	0.032	0.0073
128	0.76	1.15	0.11	0.014
256	3.22	5.05	0.32	0.078
512	15.82	23.85	0.78	0.45

Figure 4.2); however, ringing may present a problem in practice. One way to overcome this would be to instead use a discrete cosine, or sine, transform decomposition, which enforces continuity, or differentiability, respectively, at the boundary. However, we do not pursue this here.

Finally, we acknowledge that the comparisons presented here are not completely comprehensive. Tests on a wider range of data, including wavefront data generated with more accurate turbulence models, e.g., the full Navier–Stokes equations, and real Shack–Hartmann sensor data, should also be done. However, we do not pursue such a study here, as our intention is only a proof of concept. For this, we have used a testing methodology that is standard in the adaptive optics community [6, 9, 10, 19] and hence feel justified in concluding that our methods are worthy of further consideration.

5. Conclusions. We have presented a detailed discussion of the problem of wavefront, or phase, reconstruction from Shack–Hartmann wavefront gradient data. This included a derivation of a discrete, stochastic linear system relating the gradient measurements to the underlying discrete phase $\boldsymbol{\phi}$; a derivation of the minimum variance estimator for $\boldsymbol{\phi}$ given the prior probability density $\boldsymbol{\phi} \sim N(\mathbf{0}, \boldsymbol{C}_{\boldsymbol{\phi}})$; and a discussion of estimates for the covariance $\boldsymbol{C}_{\boldsymbol{\phi}}$.

Computational methods for the phase reconstruction problem were then presented. First, an efficient method for computing an accurate approximation of the minimum norm least squares solution was given; it used a Cholesky factorization with sparse reordering. Then the current gold standard for accuracy and efficiency, the multigrid preconditioned conjugate gradient method (MGPCG), was compared with two new approaches presented in this paper—the so-called least squares preconditioned conjugate gradient method (LSPCG) and the gradient denoised least squares method (GDLS). The results indicate that though MGPCG yields lower values for the relative error, the methods introduced here, and, in particular, GDLS, are more efficient and yield comparable results. The implementation of GDLS used in our comparisons utilized the efficient method for computing the minimum norm least squares solution $\boldsymbol{\phi}_{\text{MNLS}}$ mentioned above, at a cost of two sparse linear system backsolves. The denoised phase estimate $\boldsymbol{\phi}_{\text{DMNLS}}$ was then computed at a computational cost of only two discrete Fourier transforms, making GDLS a very efficient approach.

We also presented results relating what is known in the adaptive optics community as waffle mode to standard discretizations of the Laplacian operator and showed that the minimum norm least squares solution, the minimum variance solution, and the denoised least squares solutions do not contain waffle mode.

Acknowledgments. First, the referees deserve thanks for their careful reading of the paper and for their constructive commentary. The paper is much better because of their efforts. The author would also like to acknowledge the help that he received from Curt Vogel in the form of discussion and for the MATLAB code that was used for data

generation in our experiments. Also, this work would not have been possible without the support of the University of Montana through its International Faculty Exchange Program and the University of Helsinki, the host university for the exchange.

REFERENCES

[1] K. L. Baker, *Least-squares wave-front reconstruction of Shack-Hartmann sensors and shearing interferometers using multigrid techniques*, Review of Scientific Instruments, 76 (2005), 053502.

[2] J. M. Beckers, *Adaptive optics for astronomy: Principles, performance, and applications*, Annu. Rev. Astron. Astrophys., 31 (1993), pp. 13–62.

[3] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, 2000.

[4] D. L. Fried, *Least-squares fitting a wave-front distortion estimate to an array of phase-difference measurements*, J. Opt. Soc. Am., 67(3), (1977).

[5] J. W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, New York, 1996.

[6] B. L. Ellerbroek, *Efficient computation of minimum-variance wave-front reconstructors with sparse matrix techniques*, J. Opt. Soc. Amer. A, 19 (2002), pp. 1803–1816.

[7] L. Evans, *Partial Differential Equations*, AMS, Providence, RI, 1998.

[8] R. C. Flicker, *Efficient first-order performance estimation for high-order adaptive optics systems*, Astronomy and Astrophysics, 405 (2003), pp. 1177–1189.

[9] L. Gilles, *Order-N sparse minimum-variance open-loop reconstructor for extreme adaptive optics*, Optics Letters, 28 (2003), pp. 1927–1929.

[10] L. Gilles, C.R. Vogel, and B. Ellerbroek, *A multigrid preconditioned conjugate gradient method for large scale wavefront reconstruction*, J. Opt. Soc. Amer. A, 19 (2002), pp. 1817–1822.

[11] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, Fundamentals of Algorithms 3, SIAM, Philadelphia, 2006.

[12] J. Herrmann, *Least-squares wave front errors of minimum norm*, J. Opt. Soc. Amer., 70 (1980), pp. 28–35.

[13] R. H. Hudgin, *Wave-front reconstruction for compensated imaging*, J. Opt. Soc. Amer., 67 (1977), pp. 375–378.

[14] R. B. Makidon, A. Sivaramakrishnan, M. D. Perrin, L. C. Roberts, B. R. Oppenheimer, R. Soummer, J. Graham, *An analysis of fundamental waffle mode in early AEOS adaptive optics images*, Publications of the Astronomical Society of the Pacific, 117 (2005), pp. 831–846.

[15] M. D. Pritt, *Phase Unwrapping by means of multigrid techniques for interferometric SAR*, IEEE Trans. Geoscience and Remote Sensing, 34 (1996), pp. 728–738.

[16] M. Roggemann and B. Welsh, *Imaging Through Turbulence*, CRC Press, Boca Raton, FL, 1996.

[17] W. H. Southwell, *Wave-front estimation from wave-front slope measurements*, J. Opt. Soc. Amer., 70 (1980), pp. 998–1006.

[18] C. R. Vogel, *Computational Methods for Inverse Problems*, Frontiers Appl. Math. 23, SIAM, Philadelphia, 2002.

[19] C. R. Vogel and Q. Yang, *Multigrid algorithm for least-squares wavefront reconstruction*, Applied Optics, 45 (2006), pp. 705–715.

[20] B. M. Welsh, B. L. Ellerbroek, M. C. Roggeman, and T. L. Pennington, *Fundamental performance comparison of a Hartmann and a shearing interferometer wave-front sensor*, Applied Optics, 34 (1995), pp. 4186–4195.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 84–101

FUNCTIONS PRESERVING NONNEGATIVITY OF MATRICES*

GAUTAM BHARALI† AND OLGA HOLTZ‡

Abstract. The main goal of this work is to determine which entire functions preserve nonnegativity of matrices of a fixed order n—i.e., to characterize entire functions f with the property that $f(A)$ is entrywise nonnegative for every entrywise nonnegative matrix A of size $n \times n$. Toward this goal, we present a complete characterization of functions preserving nonnegativity of (block) upper-triangular matrices and those preserving nonnegativity of circulant matrices. We also derive necessary conditions and sufficient conditions for entire functions that preserve nonnegativity of symmetric matrices. We also show that some of these latter conditions characterize the even or odd functions that preserve nonnegativity of symmetric matrices.

Key words. nonnegative inverse eigenvalue problem, circulant matrices, (block) upper-triangular matrices, symmetric matrices, positive definite matrices, entire functions, divided differences

AMS subject classifications. 15A29, 15A48, 15A42

DOI. 10.1137/050645075

1. Motivation.

The purpose of this paper is to investigate which entire functions preserve nonnegativity of matrices of a fixed order. More specifically, we consider several classes of structured matrices whose structure is preserved by entire functions and characterize those entire functions f with the property that $f(A)$ is entrywise nonnegative for each entrywise nonnegative matrix A of size $n \times n$. The characterizations that we obtain might be of independent interest in matrix theory and other areas of mathematics. One of our own motivations behind our investigation is its relevance to the inverse eigenvalue problem for nonnegative matrices.

The long-standing inverse eigenvalue problem for nonnegative matrices is the problem of determining, given an n-tuple (multiset) Λ of complex numbers, whether there exists an entrywise nonnegative matrix A whose spectrum $\sigma(A)$ is Λ. The literature on the subject is vast, and we make no attempt to review it. The interested reader is referred to books [25] and [1], expository papers [3], [9], [18], [19], and references therein, as well as to some recent papers [23], [4], [21], [30], [31], [32], [20], [26].

The necessary conditions for a given n-tuple to be realizable as the spectrum of a nonnegative matrix known so far for arbitrary values of n can be divided into three groups: conditions for nonnegativity of moments, Johnson–Loewy–London inequalities, and Newton's inequalities.

Given an n-tuple Λ, its [illegible] are defined as follows:

$$s_m(\Lambda) := \sum_{\lambda \in \Lambda} \lambda^m, \qquad m \in \mathbb{N}.$$

If $\Lambda = \sigma(A)$ for some nonnegative matrix A, then $s_m(\Lambda)$ is nothing but the trace $\operatorname{tr}(A^m)$ and therefore must be nonnegative. Another basic condition follows from the

*Received by the editors November 12, 2005; accepted for publication (in revised form) by M. L. Overton June 19, 2007; published electronically February 6, 2008.
http://www.siam.org/journals/simax/30-1/64507.html

†Department of Mathematics, Indian Institute of Science, Bangalore 560012, India (bharali@math.iisc.ernet.in).

‡Department of Mathematics, University of California, Berkeley, CA 94720 (holtz@math.berkeley.edu).

Perron–Frobenius theory [27], [11]: the largest absolute value $\max_{\lambda\in\Lambda}|\lambda|$ must be the Perron eigenvalue of a realizing matrix A and therefore must itself be in Λ. Finally, the multiset Λ must be closed under complex conjugation, being the spectrum of a real matrix A. Interestingly, the last two conditions are in fact not independent conditions but follow from the nonnegativity of moments, as was shown by Friedland in [10]. Thus, there turns out to be just one set of basic conditions:

$$s_m(\Lambda) \geq 0 \qquad \text{for } m \in \mathbb{N}.$$

The next set of necessary conditions was discovered independently by Loewy and London in [22] and by Johnson in [17]. These conditions relate moments among themselves as follows:

$$s_k^m(\Lambda) \leq n^{m-1} s_{km}(\Lambda), \quad k, m \in \mathbb{N}.$$

Newton's inequalities were conjectured in [14] and proved for M-matrices in [13]. An M-matrix is a matrix of the form $rI - A$, where A is a nonnegative matrix, $r \geq \varrho(A)$, and $\varrho(A)$ is the spectral radius of A:

$$\varrho(A) := \max_{\lambda\in\sigma(A)} |\lambda|.$$

If M is an M-matrix of order n, then the normalized coefficients $c_j(M)$ of its characteristic polynomial defined by

$$\det(\lambda I - M) =: \sum_{j=0}^{n} (-1)^j \binom{n}{j} c_j(M) \lambda^{n-j}$$

must satisfy Newton's inequalities

$$c_j^2(M) \geq c_{j-1}(M) c_{j+1}(M), \quad j = 1, \ldots, n-1.$$

Since the coefficients $c_j(M)$ are determined entirely by the spectrum of M, and the latter is obtained from the spectrum of a nonnegative matrix A by an appropriate shift, Newton's inequalities form yet another set of conditions necessary for an n-tuple to be realizable as the spectrum of a nonnegative matrix. The above three sets of conditions, i.e., nonnegativity of moments, Johnson–Loewy–London inequalities, and Newton's inequalities, are all independent of each other but are not sufficient for realizability of a given n-tuple (see [13]).

Quite a few sufficient conditions are also known (see, e.g., [37], [19], [10], [3]) as well as certain techniques for perturbing or combining realizable n-tuples into new realizable n- or m-tuples (where $m \geq n$) (see, e.g., [34], [33], [31]). Also, necessary and sufficient conditions on an n-tuple to serve as the nonzero part of the spectrum of some nonnegative matrix are due to Boyle and Handelman [2].

Finally, it follows from the Tarski–Seidenberg theorem [38, 29] that all realizable n-tuples form a semialgebraic set (see also [16]); i.e., for any given n, there exist only finitely many polynomial inequalities that are necessary and sufficient for an n-tuple Λ to be realizable as the spectrum of some nonnegative matrix A (this observation was communicated to us by Friedland).

Indeed, each realizable n-tuple $\Lambda = (\lambda_1, \ldots, \lambda_n)$ is characterized by the condition

$$\exists\, A \geq 0 \;:\; \det(\lambda I - A) = \prod_{j=1}^{n} (\lambda - \lambda_j).$$

The last condition is equivalent to each elementary symmetric function $\sigma_j(\Lambda)$ being equal to the jth coefficient of the characteristic polynomial of A multiplied by $(-1)^j$, i.e., to the sum of all principal minors of A of order j, for $j = 1, \ldots, n$. Since the set of all nonnegative matrices is a semialgebraic set in n^2 entries of the matrix and since each sum of all principal minors of A of order j is a polynomial in the entries of A, the lists of coefficients of characteristic polynomials of nonnegative matrices form a semialgebraic set, and hence the n-tuples whose elementary symmetric functions match one of those lists also form a semialgebraic set by the Tarski–Seidenberg theorem.

However, despite so many insights into the subject, and despite the results obtained so far, the nonnegative inverse eigenvalue problem remains open. In fact, the problem remains open when specialized to several important classes of structured matrices—for instance, the class of entrywise nonnegative symmetric matrices.

Note that the three sets of conditions on an n-tuple Λ that we discussed above, i.e., nonnegativity of moments, the Johnson–Loewy–London inequalities, and the Newton inequalities, are necessary conditions for the realizability of Λ as the spectrum of a $n \times n$ matrix with nonnegative entries (provided, of course, that all the entries of Λ are now real). A significant fraction of this paper will be devoted to an idea that has relevance to the inverse eigenvalue problem for nonnegative symmetric matrices. It is an idea that was first expressed by Loewy and London in [22]. When adapted to symmetric matrices, it may be stated as follows: Suppose a primary matrix function f is known to map nonnegative symmetric matrices of some fixed order n into themselves. Thus $f(A)$ is nonnegative whenever A is. Since $f(\sigma(A)) = \sigma(f(A))$, both the spectrum $\sigma(A)$ and its image under the map f must then satisfy the aforementioned conditions for realizability. This enlarges the class of necessary conditions for the symmetric nonnegative inverse eigenvalue problem. Describing this larger class would require knowing exactly what functions f preserve nonnegativity of such matrices (of a fixed order). Toward this end, we provide a characterization of all the and entire functions that preserve entrywise nonnegativity of nonnegative symmetric matrices.

Along the way, we also obtain complete characterizations of all entire functions that preserve nonnegativity of the following classes of structured matrices:

- triangular and block-triangular matrices and
- circulant matrices.

We ought to add here that, for the above classes of structured matrices, our results do not have a bearing on the nonnegative inverse eigenvalue problems associated to them. In fact, the solutions of the latter problems are quite straightforward. To be precise, an n-tuple Λ is the spectrum of an $n \times n$ triangular matrix if and only if all the entries of Λ are nonnegative. As for circulants, the eigenvalues of a circulant matrix A are determined by its first row $\mathbf{a} := [a_0\ a_1 \ldots a_{n-1}]$ (see [7]), and, in fact, there is a constant matrix W (i.e., independent of $\mathbf{a}$ and A) such that $\sigma(A) = \mathbf{a}\mathsf{W}$. Thus the realizable n-tuples in this case are of the form $\mathbf{a}\mathsf{W}$, $\mathbf{a} \in \mathbb{R}^n_+$. Nevertheless, we feel that the problem of characterizing the functions that preserve nonnegativity of the above classes of matrices can be of interest, independently of the nonnegative inverse eigenvalue problem.

2. Outline. This paper is organized as follows. We make several preliminary observations in section 4. Before focusing our attention on aspects of the symmetric nonnegative inverse eigenvalue problem, we study the structured matrices just discussed. In section 5, we characterize the class of functions preserving nonnegativity of triangular and block triangular matrices. It turns out that these are characterized

by nonnegativity conditions on their divided differences over the nonnegative reals. Next, in section 6, we obtain a characterization of functions preserving nonnegativity of circulant matrices. This characterization is quite different from that in section 5; it involves linear combinations of function values taken at certain nonreal points of $\mathbb{C}$. In section 7, we obtain a complete characterization of the class $\mathcal{F}_n$ for small values of n.

The remainder of the paper is essentially devoted to functions that preserve nonnegativity of symmetric matrices. In section 8.1, we review existing results in that direction. In particular, we discuss the restriction of [24, Corollary 3.1] to entire functions, which claims to provide a characterization of entire functions that preserve entrywise nonnegativity of symmetric matrices of a fixed order. We point out that, while this result is true when restricted to nonnegative symmetric matrices, the condition occurring in that result is for an entire function to preserve nonnegativity of all symmetric matrices. The leading to [24, Corollary 3.1], however, turn out to be very useful. We use these techniques, along with some new ideas, to obtain necessary conditions and sufficient conditions and characterizations of the and entire functions that preserve nonnegativity of symmetric matrices of a fixed order. This is the content of sections 8.2 and 8.3. Because of a gap between the necessary and the sufficient conditions, which we also point out in section 8.2, the results of that section do not provide a characterization of functions preserving nonnegativity of symmetric matrices. We end the paper with a list of several open problems in section 9 and suggest various approaches to their solution that we have not explored in this paper.

3. Notation. We use standard notation $\mathbb{R}^{m\times n}$ for real matrices of size $m\times n$, $\mathbb{R}_+$ for nonnegative reals, $\mathbb{Z}_+$ for nonnegative integers, $A \geq 0$ ($A > 0$) to denote that a matrix A is entrywise nonnegative (positive), and $\sigma(A)$ to denote the spectrum of A. For $x \in \mathbb{R}$, we use $\lfloor x \rfloor$ to denote the greatest integer that is less than or equal to x.

4. Preliminaries. The main goal of the paper is to characterize functions f such that the matrix $f(A)$ is (entrywise) nonnegative for any nonnegative matrix A of order n. Since the primary matrix function $f(A)$ is defined in accordance with values of f and its derivatives on the spectrum of A (see, e.g., [15, sections 6.1, 6.2]), we want to avoid functions that are not differentiable at some points in $\mathbb{C}$. Therefore, we restrict ourselves to functions that are analytic everywhere in $\mathbb{C}$, i.e., to entire functions. Thus we consider the class

$$\mathcal{F}_n := \{f \text{ entire} : A \in \mathbb{R}^{n\times n},\ A \geq 0 \implies f(A) \geq 0\}.$$

Note right away that the classes $\mathcal{F}_n$ are ordered by inclusion.

LEMMA 1. $n \in \mathbb{N}$, $\mathcal{F}_n \supseteq \mathcal{F}_{n+1}$

. Let A be a nonnegative matrix of order n, and let $f \in \mathcal{F}_{n+1}$. Consider the block diagonal matrix $B := \operatorname{diag}(A, 0)$ obtained by adding an extra zero row and column to A. Since $f(B) = \operatorname{diag}(f(A), 0)$, the matrix $f(A)$ must be nonnegative. Thus $f \in \mathcal{F}_n$. □

Recall that any entire function can be expanded into its Taylor series around any point in $\mathbb{C}$ and that the resulting series converges everywhere (see, e.g., [5]). We will mostly focus on Taylor series of functions in $\mathcal{F}_n$ centered at the origin. We start with some simple observations regarding a few initial Taylor coefficients of such a function.

PROPOSITION 2. $f(z) = \sum_{j=0}^{\infty} a_j z^j$ $\mathcal{F}_n$, $a_j \geq 0$ $j = 0, \ldots, n-1$

. For $n = 1$, the statement follows from evaluating f at 0. If $n > 1$ and $f \in \mathcal{F}_n$, then evaluate the function f at the matrix

$$A := \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Since

$$f(A) = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-2} & a_{n-1} \\ 0 & a_0 & a_1 & \cdots & a_{n-3} & a_{n-2} \\ 0 & 0 & a_0 & \cdots & a_{n-4} & a_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_0 & a_1 \\ 0 & 0 & 0 & \cdots & 0 & a_0 \end{bmatrix},$$

the entries $a_0, \ldots, a_{n-1}$ of $f(A)$ must be nonnegative. This finishes the proof. □

COROLLARY 3. f $\mathcal{F}_n$ $n \in \mathbb{N}$ $f(z) = \sum_{j=0}^{\infty} a_j z^j$ $a_j \geq 0$ $j \in \mathbb{Z}_+$

. One direction follows from Proposition 2. The other direction is trivial: if all terms in the Taylor expansion of f around the origin are nonnegative, then $f(A)$ combines powers of a nonnegative matrix A using nonnegative coefficients, so the resulting matrix is nonnegative. Here we make use of the standard fact [15, Theorem 6.2.8] that the matrix power series $\sum_{j=0}^{\infty} a_j A^j$ converges to $f(A)$. □

. It must be noted that Proposition 2 be a necessary condition for an entire function to belong to $\mathcal{F}_n$. This is easy to see; fix an $n \in \mathbb{N}$ and set

$$F(x) = -x^n + \sum_{j=0}^{n-1} a_j x^j,$$

where we choose $a_j \geq 0$, $j = 0, \ldots, n-1$. Then, there exists an $x_0 > 0$ such that $F(x) < 0$ for all $x \in (x_0, \infty)$. If we set $A = rI$ for some $r \in (x_0, \infty)$, then A is entrywise nonnegative, while the diagonal entries of $F(A)$ are negative. Hence, although $a_j \geq 0$ for $j = 0, \ldots, n-1$, F does not preserve nonnegativity.

To conclude this section, we make two more general observations.

LEMMA 4. f $\mathcal{F}_n$ n

. This is simply due to the continuity of f, since the set of strictly positive matrices is dense in the set of all nonnegative matrices of order n. □

LEMMA 5. f, P, D , $f(A)$ $f($ $)^{-1})$

. Note that $($ $)f(A)($ $)^{-1} = f($ $($ $)^{-1})$ and that both matrices PD and $($ $)^{-1}$ are nonnegative. So, $f(A)$ is nonnegative if and only if the matrix $f($ $($ $)^{-1})$ is nonnegative. □

We now analyze three superclasses of our class $\mathcal{F}_n$:

- entire functions preserving nonnegativity of upper-triangular matrices,
- entire functions preserving nonnegativity of circulant matrices, and
- entire functions preserving nonnegativity of symmetric matrices.

5. Preserving nonnegativity of (block) triangular matrices. We first discuss functions preserving nonnegativity of upper- (or lower-)triangular matrices. The characterization that we obtain makes use of the notion of divided differences. The [illegible] (see, e.g., [8]) of a smooth function f at points $x_1, \dots, x_k$ (which can be thought of as an ordered sequence $x_1 \le \cdots \le x_k$) is usually defined via the recurrence relation

$$f[x_1, \dots, x_k] := \begin{cases} \frac{f[x_2,\dots,x_k]-f[x_1,\dots,x_{k-1}]}{x_k-x_1}, & x_1 \neq x_k, \\ f^{(k-1)}(x_1)/(k-1)!, & x_1 = x_k, \end{cases}$$

and where $f[x] := f(x)$. Divided differences play a large part in this paper. We shall, however, make no attempt to review the results on divided differences that we shall draw upon, especially since they are quite readily accessible. The interested reader is referred to [8].

THEOREM 6. [illegible] f [illegible] n [illegible] n [illegible] $\mathbb{R}_+$ [illegible],

$$f[x_1, \dots, x_k] \ge 0 \quad \text{for } x_1, \dots, x_k \ge 0, \quad k = 1, \dots, n, \tag{1}$$

[illegible] f [illegible] $n-1$ [illegible] $\mathbb{R}_+$ [illegible] Let $A =: (a_{ij})$ be a nonnegative upper-triangular matrix. Suppose a function f satisfies (1). By [28], [35] (see also [36]), the elements of the matrix $f(A)$ can be written explicitly as

$$f(A)_{ij} = \begin{cases} f(a_{ii}), & i = j, \\ \sum_{i<i_1<\cdots<i_k<j} a_{ii_1} \cdots a_{i_kj} f[a_{ii}, a_{i_1i_1}, \dots, a_{i_ki_k}, a_{jj}], & i < j, \\ 0, & i > j. \end{cases} \tag{2}$$

The divided differences appearing in the sum on the right-hand side are of order not exceeding n; hence all the summands, and therefore the sums, are nonnegative.

[illegible] We proceed by induction on n. If f preserves nonnegativity of upper-triangular matrices of order n, it does so also for matrices of order $n-1$. Thus, by our inductive hypothesis, (1) holds up to order $n-1$. To see that all divided differences of order n are also nonnegative over nonnegative reals, consider the matrix A whose first upper diagonal consists of ones, whose main diagonal consists of n arbitrary nonnegative numbers $x_1, \dots, x_n$, and whose other entries are zero. Then, (2) shows that $f(A)_{1n} = f[x_1, \dots, x_n]$ and must be nonnegative.

Finally, since all divided differences of a fixed order k at points in a domain D are nonnegative if and only if $f^{(k-1)}(x)$ is nonnegative for every point $x \in D$ [8], we see that condition (1) is equivalent to all derivatives of f of order up to $n-1$ being nonnegative on $\mathbb{R}_+$. This finishes the proof. □

The proofs of (2) in [35] and [28] are based on the following observation.

RESULT 7 (see [35], [28]). [illegible]

$$M = \begin{bmatrix} A & B \\ 0 & a \end{bmatrix}, \qquad a \in \mathbb{C} \setminus \sigma(A),$$

$$f(M) = \begin{bmatrix} f(A) & (A-aI)^{-1}(f(A)-f(a)I)B \\ 0 & f(a) \end{bmatrix}$$

f

One can prove an analogous statement in the block triangular case.

PROPOSITION 8. f

$$M = \begin{bmatrix} A & B \\ 0 & C \end{bmatrix}, \qquad \sigma(A) \cap \sigma(C) = \emptyset.$$

$$f(M) = \begin{bmatrix} f(A) & f(A)X - Xf(C) \\ 0 & f(C) \end{bmatrix},$$

X $(\quad)$

$$AX - XC = B.$$

. Let X be a solution of the Sylvester equation $AX - XC = B$. Since the spectra of A and C are disjoint, this solution is unique [15, section 4.4]. Then, $M = T^{-1}\operatorname{diag}(A, C)T$, where

$$T = \begin{bmatrix} I & X \\ 0 & I \end{bmatrix}.$$

Hence $f(M) = T^{-1}\operatorname{diag}(f(A), f(C))T$, which proves the proposition. □

As an immediate corollary, we obtain an indirect characterization of functions preserving nonnegativity of block triangular matrices with two diagonal blocks.

COROLLARY 9. f

$$\begin{bmatrix} A & B \\ 0 & C \end{bmatrix}, \qquad A \in \mathbb{R}^{n_1 \times n_1},\ C \in \mathbb{R}^{n_2 \times n_2},$$

(a) $f \in \mathcal{F}_N$, $N := \max\{n_1, n_2\}$;

(b) $f(A)X - Xf(C) \geq 0$ $A \in \mathbb{R}^{n_1 \times n_1}$, $B \in \mathbb{R}^{n_1 \times n_2}$, $C \in \mathbb{R}^{n_2 \times n_2}$ $A, B, C \geq 0$, $\sigma(A) \cap \sigma(C) = \emptyset$ $(\quad)$ X $AX - XC = B$

. For f to preserve nonnegativity of blocks A and C, it has to belong to $\mathcal{F}_N$ (keeping in mind Lemma 1). The remainder of our assertion follows from Proposition 8 and the fact that the matrices with nonnegative blocks A, B, C, such that the spectra of A and C are disjoint, are dense in the set of all block upper-triangular matrices. □

The above proposition, however, does not allow for an explicit formula of the type (1) as in Theorem 6.

. Note that the results of this section characterize functions preserving nonnegativity of the (block) lower-triangular matrices as well.

6. Preserving nonnegativity of circulant matrices. A [illegible] (see, e.g., [7]) A is determined by its first row $(a_0, \dots, a_{n-1})$ as follows:

$$\begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \\ a_{n-1} & a_0 & a_1 & \cdots & a_{n-2} \\ a_{n-2} & a_{n-1} & a_0 & \cdots & a_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & a_3 & \cdots & a_0 \end{bmatrix}.$$

All circulant matrices of size n are polynomials in the basic circulant matrix

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix},$$

which implies in particular that any function $f(A)$ of a circulant matrix is a circulant matrix as well. Moreover, the eigenvalues of a circulant matrix are determined by its first row (see [7]) by the formula

$$\left\{ \sum_{j=0}^{n-1} \omega^{kj} a_j \ : \ k = 0, \dots, n-1 \right\}, \quad \text{where } \omega := e^{2\pi \mathrm{i}/n}.$$

Hence the eigenvalues of $f(A)$ are

$$\left\{ f\left(\sum_{j=0}^{n-1} \omega^{kj} a_j \right) \ : \ k = 0, \dots, n-1 \right\}.$$

Thus, the elements $(f_0, \dots, f_{n-1})$ of the first row of $f(A)$ can be read off from its spectrum:

$$f_l = \frac{1}{n} \sum_{k=0}^{n-1} \omega^{-lk} f\left(\sum_{j=0}^{n-1} \omega^{jk} a_j \right), \quad l = 0, \dots, n-1.$$

This argument proves the following theorem.

THEOREM 10. [illegible] f [illegible] n, [illegible] $l = 0, \dots, n-1$,

$$\sum_{k=0}^{n-1} \omega^{-lk} f\left(\sum_{j=0}^{n-1} \omega^{jk} a_j \right) \geq 0 \quad \textit{whenever } a_j \geq 0, \ j = 0, \dots, n-1, \tag{3}$$

$$\textit{where } \omega = e^{2\pi \mathrm{i}/n}.$$

7. Characterization of $\mathcal{F}_n$ for small values of n. We now focus on the function classes $\mathcal{F}_n$ for small values of n. Recall the inclusion $\mathcal{F}_{n+1} \subseteq \mathcal{F}_n$ from Lemma 1, which means that all conditions satisfied by the functions from $\mathcal{F}_n$ get inherited by the functions from $\mathcal{F}_{n+1}$. Thus we need to find out precisely how to strengthen the conditions that determine $\mathcal{F}_n$ to get to the next class $\mathcal{F}_{n+1}$.

7.1. The case $n = 1$. A function f is in $\mathcal{F}_1$ if and only if f maps nonnegative reals into themselves. While this statement is in a way a characterization in itself, if f is an entire function with , we can give a description of the form that f takes. For such f, the proposition below serves as an alternative characterization.

PROPOSITION 11. [illegible] f [illegible] $\mathcal{F}_1$ [illegible]

$$f(z) = g(z) \prod_{\alpha,\beta} ((z+\alpha)^2 + \beta^2) \prod_{\gamma} (z+\gamma), \tag{4}$$

[illegible] α, [illegible] β, [illegible], [illegible] γ, [illegible], [illegible] g [illegible] $\mathbb{C}$ [illegible] $\mathbb{R}_+$ [illegible].

PROOF. First note that since f takes real values over the nonnegative reals, all its zeros occur in conjugate pairs. Moreover, while the multiplicity of the real negative zeros is not restricted in any way, the nonnegative zeros must occur with even multiplicities. This produces exactly the factors recorded in (4), with nonnegative zeros corresponding to $\beta = 0$. After factoring out all the linear factors, we are left with an entire function—which we call $g(z)$—that has no zeros and takes only positive values on $\mathbb{R}_+$. This gives us the expression (4). □

REMARK. Incidentally, all polynomials f that take only positive values on $\mathbb{R}_+$ are characterized by a theorem due to Poincaré and Pólya (see, e.g., [6, p. 175]): there exists a number $N \in \mathbb{Z}_+$ such that the polynomial $(1+z)^N f(z)$ must have positive coefficients. Since we include nonpolynomial functions in our class $\mathcal{F}_1$, and since we allow functions to have zeros in $\mathbb{R}_+$, the Poincaré–Pólya characterization is not directly relevant to our setup.

7.2. The case $n = 2$. We just saw that functions in $\mathcal{F}_1$ are characterized by one inequality, viz.

$$f(x) \geq 0 \quad \forall\, x \geq 0. \tag{5}$$

In this subsection we will see that functions in $\mathcal{F}_2$ are characterized by two inequalities, one involving a divided difference. We recall two preliminary observations, Lemmas 4 and 5 that were proved in section 4. Their specialization to the case $n = 2$ gives the following corollary.

COROLLARY 12. [illegible] f [illegible] $\mathcal{F}_2$ [illegible] 2 [illegible]

PROOF. A strictly positive 2×2 matrix A can be symmetrized by using the transformation DAD^{-1}, where D is a diagonal matrix with positive diagonal elements. Thus, Lemmas 4 and 5 imply that $f(A)$ is nonnegative for all strictly positive, and hence for all nonnegative, matrices A of order 2 if and only if $f(A)$ is nonnegative for all symmetric matrices. □

Now we are in a position to prove a characterization theorem for the class $\mathcal{F}_2$.

THEOREM 13. [illegible] f [illegible] $\mathcal{F}_2$ [illegible]

$$f(x+y) - f(x-y) \geq 0 \quad \forall\, x, y \geq 0, \tag{6}$$

$$(7) \qquad (x+y-z)f(x-y)+(z-x+y)f(x+y)\geq 0 \quad \forall\, x\geq z\geq 0,\ y\geq x-z,$$

[illegible] f [illegible] (6) [illegible]

$$(8) \qquad (x+y)f(x-y)+(y-x)f(x+y)\geq 0 \quad \forall\, y\geq x\geq 0.$$

[illegible]. If $f\in\mathcal{F}_2$, then, in particular, f preserves nonnegativity of nonnegative circulant matrices. Thus, the conditions (3) are necessary for f to belong to $\mathcal{F}_2$. Observe that the condition (6) is one of the two necessary conditions (3) in case $n=2$ (taking $a_0=x$ and $a_1=y$). Therefore, we need to check that the condition (7) is also necessary and that both together are sufficient. Then we also need to check that conditions (6) and (7) are equivalent to conditions (6) and (8).

By Corollary 12, we can restrict ourselves to the case when A is a positive symmetric matrix, i.e., when

$$A=\begin{bmatrix} a_{11} & b \\ b & a_{22}\end{bmatrix}, \quad a_{11},b,a_{22}>0.$$

Since the value of f at A coincides with the value of its interpolating polynomial of degree 1 with nodes of interpolation chosen at the eigenvalues of A [15, sections 6.1, 6.2], we get

$$f(A)=f[r_1]I+f[r_1,r_2](A-r_1I),$$

where

$$r_j:=\frac{a_{11}+a_{22}}{2}+(-1)^j\frac{\sqrt{(a_{11}-a_{22})^2+4b^2}}{2}, \quad j=1,2.$$

So, the off-diagonal entries of $f(A)$ are equal to

$$f[r_1,r_2]b,$$

while the diagonal entries are

$$f[r_1,r_2](a_{jj}-r_1)+f(r_1), \quad j=1,2.$$

Writing

$$x:=\frac{a_{11}+a_{22}}{2},$$

$$y:=\frac{\sqrt{(a_{11}-a_{22})^2+4b^2}}{2},$$

$$z:=\min(a_{11},a_{22}),$$

we see that the characterization for $\mathcal{F}_2$ consists precisely of conditions (6) and (7).

It remains to prove that (6) and (7) are equivalent to (6) and (8). By simply taking $z=0$ in (7), we see that (7) implies (8). So let us now assume (6) and (8). We begin by stating a simple auxiliary fact. Taking $x=0$ and $y>0$ in (6) and (8), we get $f(y)\pm f(-y)\geq 0$ for all $y>0$. We conclude from this that $f(y)\geq 0$ whenever $y\geq 0$, i.e., that f satisfies (5).

First consider y lying in the range $x - z \leq y \leq x$. In this case, we get

$$(x+y-z)f(x-y) + (z-x+y)f(x+y)$$
$$= (y-(x-z))(f(x+y) - f(x-y)) + 2yf(x-y) \geq 0 \qquad \text{for } \ x \geq z \geq 0.$$

The nonnegativity of the second term above is a consequence of (5), since $x - y$ is nonnegative in this case. Now if $y \geq x$, then (6) and (8) simply imply that

$$(x+y-z)f(x-y) + (z-x+y)f(x+y)$$
$$= ((x+y)f(x-y) + (y-x)f(x+y)) + z(f(x+y) - f(x-y)) \geq 0$$
$$\text{for } \ y \geq x \geq z \geq 0.$$

The last two inequalities show that (6) and (8) imply (6) and (7). This finishes the proof. □

8. Preserving nonnegative symmetric matrices. We now focus on the characterization problem for the class of entire functions which preserve nonnegativity of symmetric matrices. We begin by recalling known facts about functions which preserve nonnegative symmetric matrices that are in addition nonnegative definite, i.e., have only nonnegative eigenvalues.

8.1. Preserving nonnegative definite nonnegative symmetric matrices. Interestingly, the condition necessary and sufficient for preserving nonnegative symmetric matrices that are nonnegative definite turns out to be exactly the same as the condition for preserving upper- (or lower-)triangular nonnegative matrices.

The characterization of functions which preserve the class of [illegible], entrywise nonnegative symmetric matrices is due to Micchelli and Willoughby [24]. We next state a version of their result that is useful for our purposes.

RESULT 14 (version of [24, Corollary 3.1]). [illegible] f [illegible] f [illegible] n [illegible] $\mathbb{R}_+$, [illegible] f [illegible] (1) [illegible] $f^{(j)}$ [illegible] f [illegible] $n-1$ [illegible] $\mathbb{R}_+$

The proof of Result 14 in [24] relies on two facts. The first is that $f(A)$ coincides with the interpolating polynomial of f, with nodes at the eigenvalues of A, evaluated at A, i.e., that

$$f(A) = f[r_1]I + f[r_1, r_2](A - r_1 I) + \cdots + f[r_1, \ldots, r_n](A - r_1 I)\cdots(A - r_{n-1}A). \tag{9}$$

The second fact is the entrywise nonnegativity of all matrix products

$$(A - r_1 I)\cdots(A - r_j I), \qquad j = 1, \ldots, n-1,$$

which holds under the assumption that the eigenvalues $r_1, \ldots r_n$ of A are ordered

$$r_1 \leq r_2 \leq \cdots \leq r_n.$$

Observe, however, that condition (1) is not sufficient for a function to preserve nonnegativity of [illegible] nonnegative symmetric matrices. Indeed, let $n = 2$, and let

$$f(x) = 1 + x + \frac{1}{2}x^2 - \frac{2}{3}x^3 + \frac{1}{4}x^4.$$

This function satisfies the condition (1) with $n = 2$, but it maps the matrix

$$\begin{bmatrix} 0 & M \\ M & 0 \end{bmatrix},$$

which is [illegible], to a matrix with negative off-diagonal entries when $M > 0$ is chosen to be sufficiently large. In fact, any $M > \sqrt{3/2}$ will produce a matrix with negative entries.

Motivated by Result 14, we would therefore like to find out what conditions are necessary and sufficient for a function to preserve nonnegativity of nonnegative symmetric matrices. We begin, in the next subsection, by analyzing even and odd functions.

8.2. Even and odd functions preserving nonnegativity of symmetric matrices. Using the Micchelli–Willoughby result, i.e., Result 14 from the previous section, and an auxiliary result from [12], we shall obtain a characterization of even and odd functions which preserve nonnegativity of symmetric matrices. Our proof below will require the notion of a Jacobi matrix and that of a symmetric antibidiagonal matrix. A [illegible] is a real, nonnegative definite, tridiagonal symmetric matrix having positive subdiagonal entries. A matrix A is called a [illegible] if it has the form

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & a_n \\ 0 & 0 & \cdots & a_{n-2} & a_{n-1} \\ \vdots & \vdots & . & \vdots & \vdots \\ 0 & a_{n-2} & \cdots & 0 & 0 \\ a_n & a_{n-1} & \cdots & 0 & 0 \end{bmatrix}, \quad a_1, \ldots, a_n \in \mathbb{R}. \tag{10}$$

We make use of the next two results from [24] and from [12].

RESULT 15 (see [24]). [illegible] f, [illegible] n, [illegible] n, [illegible] f, [illegible] n, (1) [illegible] n, $x_1 \le x_2 \le \cdots \le x_n$ [illegible].

The above result is not stated in precisely these words in [24], but it is easily inferred—it lies at the heart of the proof of [24, Theorem 2.2]. In addition, we shall also need the following result.

RESULT 16 (Corollary 3 in [12]). [illegible] $\mathcal{M}$ [illegible] n, [illegible] $\mathcal{M}$, (10) [illegible] a_j [illegible].

We are now in a position obtain a characterization of even and odd matrix functions that are of interest to us.

THEOREM 17. [illegible] $f(z) =: g(z^2)$, [illegible] n, [illegible] g, [illegible] n $\mathbb{R}_+$, [illegible] g [illegible] (1). [illegible] $f(z) =: zh(z^2)$, [illegible] n, [illegible] h [illegible] (1).

[illegible]. Let f be even. Then, $f(z) = g(z^2)$ for some entire function g. If a matrix A is entrywise nonnegative symmetric, then A^2 is entrywise nonnegative, symmetric, and nonnegative definite. By Result 14, if g satisfies (1), then $g(A^2)$ is nonnegative.

To prove the converse, consider an arbitrary n-tuple $\mathcal{M}$ of positive numbers. We can think of $\mathcal{M}$ as being ordered

$$\mathcal{M} = (x_1, \dots, x_n), \qquad x_1 \le \cdots \le x_n. \tag{11}$$

By Result 16, there exists a nonnegative symmetric antibidiagonal matrix A such that A^2 is a Jacobi matrix with spectrum $\mathcal{M}$. Then, by Result 15, the divided differences of g must be nonnegative when evaluated at the first k points of $\mathcal{M}$ for each $k = 1, \dots, n$. This implies, by the standard density reasoning, that all divided differences of g must be nonnegative over $\mathbb{R}_+$.

Now let f be odd. Then, $f(z) = zh(z^2)$ for some entire function h. If all the divided differences of h up to order n are nonnegative, then by the same argument as above, $h(A^2)$ is nonnegative for each symmetric nonnegative matrix A, and multiplication of $h(A^2)$ by a nonnegative matrix A produces a nonnegative matrix again. To prove the converse, we use induction and a technique from [24]. Since f has to preserve nonnegativity of symmetric matrices of order $n-1$ as well, we can assume the nonnegativity of the divided differences of orders $k = 1, \dots, n-1$. To prove that the nth divided difference is nonnegative, let $\mathcal{M}$ be an arbitrary positive n-tuple (11). As above, by Result 16, there exists a symmetric antibidiagonal matrix A such that A^2 is a Jacobi matrix with spectrum $\mathcal{M}$. By [24], formula (9) shows that the $(1, n)$ entry of the function $h(A^2)$ is a positive multiple of $f[x_1, \dots, x_n]$, and hence the $(1, 1)$ entry of the product $h(A^2)A$ is again a positive multiple of $f[x_1, \dots, x_n]$. Thus the nth divided difference has to be nonnegative as well, which finishes the proof. □

This theorem provides a rather natural characterization of even and odd functions that preserve nonnegativity of symmetric matrices in terms of their divided differences. However, the "natural" idea, that the even and odd parts of any entire function that preserves nonnegativity of symmetric functions must be also nonnegativity-preserving, turns out to be wrong. Here is an example that illustrates why that may not be the case.

18. Let

$$f(z) := \alpha + \beta z - z^3 + z^5 + \gamma z^6,$$

where $\beta > 1/4$ and $\alpha, \gamma > 0$ are chosen to be so large that $f(x) \ge 0$ for all $x \in \mathbb{R}$ and $f'(x) \ge 0$ for all $x \in \mathbb{R}_+$. Then, f preserves nonnegativity of symmetric matrices of order 2, but its odd part f_{odd} does not.

. The function f satisfies conditions (6) and (8). Indeed, since $f \ge 0$ on $\mathbb{R}$, we have

$$(t+s)f(-t) + tf(t+s) \ge 0 \qquad \forall\, s, t \ge 0,$$

which is equivalent to condition (8). Now, the odd part of f is given by

$$f_{odd}(z) = \beta z - z^3 + z^5 =: zh(z^2).$$

Since $\beta > 1/4$, $h(x) > 0$ for all $x \in \mathbb{R}$. Since f is monotone increasing on $\mathbb{R}_+$, we have

$$f(s+t) - f(-s) \ge f(s) - f(-s) = 2f_{odd}(s) \ge 0 \qquad \forall\, s, t \ge 0,$$

which yields condition (6). Thus, by Theorem 13, f preserves nonnegativity of symmetric matrices of order 2. However,

$$h'(x) = 2x - 1 < 0 \qquad \text{for } x < 1/2.$$

Therefore, by Theorem 17, f_{odd} does not preserve nonnegativity of symmetric functions of order 2. □

We conclude this section with a simple observation about even and odd parts of a nonnegativity-preserving function.

PROPOSITION 19. [illegible] f [illegible] n, [illegible] f_{odd} - f_{even} [illegible] $\lfloor n/2 \rfloor$

[illegible]. For n even, consider matrices of the form

$$A = \begin{bmatrix} 0 & B \\ B & 0 \end{bmatrix},$$

and for n odd, matrices of the form

$$A = \operatorname{diag}\left(\begin{bmatrix} 0 & B \\ B & 0 \end{bmatrix}, 0 \right),$$

where B is an $\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor$ symmetric nonnegative matrix. Since

$$f(A) = \begin{bmatrix} f_{even}(B) & f_{odd}(B) \\ f_{odd}(B) & f_{even}(B) \end{bmatrix} \qquad \text{for } n \text{ even},$$

$$f(A) = \operatorname{diag}\left(\begin{bmatrix} f_{even}(B) & f_{odd}(B) \\ f_{odd}(B) & f_{even}(B) \end{bmatrix}, 0 \right) \qquad \text{for } n \text{ odd},$$

we see that f_{even} and f_{odd} must preserve nonnegativity of symmetric functions of order $\lfloor n/2 \rfloor$. □

8.3. Other necessary conditions. Results from [12] allow us to derive an additional set of necessary conditions. The motivation behind these conditions is as follows. We believe that the power of Results 15 and 16—or rather, [illegible] behind those results—have not been exhausted by Theorem 17. Our next theorem is presented as an illustration of this viewpoint. On comparison with Theorem 13, we find that the conditions derived in our next theorem constitute a complete characterization for the functions of interest in the $n = 2$ case. To derive these new necessary conditions, we will need the following two results.

RESULT 20 (Theorem 1 in [12]). [illegible] Λ [illegible] (10) [illegible] a_j [illegible] $\Lambda = (\lambda_1, \ldots, \lambda_n)$ [illegible]

$$\lambda_1 > -\lambda_2 > \lambda_3 > \cdots > (-1)^{n-1}\lambda_n > 0.$$

LEMMA 21. [illegible] A [illegible] n, [illegible] A^p_{ij} [illegible] (i,j) [illegible] A^p [illegible]:

(a) [illegible] (i,j) [illegible] A^{2q-1} [illegible] $2 \le i + j \le (n - q + 1)$, $q \ge 1$

(b) [illegible] (i,j) [illegible] A^{2q} [illegible] $1 + q \le j - i \le n - 1$, $q \ge 1$

(c) [illegible] (10) [illegible] A,

$$A^{2q-1}_{1,n-q+1} = a_n a_{n-1} \ldots a_{n-2q+2}, \qquad 1 \le q \le \lfloor (n+1)/2 \rfloor, \tag{12}$$

$$A^{2q}_{1,1+q} = a_n a_{n-1} \ldots a_{n-2q+1}, \qquad 1 \le q \le \lfloor n/2 \rfloor. \tag{13}$$

. We proceed by induction on q. Note that (a), (b), and (c) are obvious when $q = 1$. Let us now assume that (a) and (b) are true for some $q < n - 3$. Note that since A is antibidiagonal,

$$A_{ij}^{2q+1} = A_{i,n-j+1}^{2q} A_{n-j+1,j} + A_{i,n-j+2}^{2q} A_{n-j+2,j}. \tag{14}$$

However, if $i + j \leq (n - (q + 1) + 1)$, then

$$(n - j + 2) - i \geq (n - j + 1) - i \geq q + 1.$$

Applying our inductive hypothesis on (b), we conclude from the above inequalities that the right-hand side of (14) reduces to zero when $i + j \leq (n - (q + 1) + 1)$. Thus, (a) is established for $q + 1$.

We establish (b) for $q + 1$ in a similar fashion. We note that

$$A_{ij}^{2q+2} = A_{i,n-j+1}^{2q+1} A_{n-j+1,j} + A_{i,n-j+2}^{2q+1} A_{n-j+2,j}. \tag{15}$$

When $j - i \geq 1 + (q + 1)$, then

$$i + (n - j + 1) \leq i + (n - j + 2) \leq n - (q + 1) + 1.$$

Since we just established (a) for $q+1$, the above inequalities tell us that the right-hand side of (15) reduces to zero when $j - i \geq 1 + (q + 1)$. Thus, (b) too is established for $q + 1$. By induction, (a) and (b) are true for all relevant q.

Part (c) now follows easily by substituting $i = 1$ and $j = n - q$ into (14) to carry out the inductive step for (12), and by substituting $i = 1$ and $j = q + 2$ into (15) to carry out the inductive step for (13). □

We can now present the aforementioned necessary conditions.

THEOREM 22. , n, $n \geq 2$, , $(x_1, \ldots, x_n)$,

$$x_1 > -x_2 > x_3 > \cdots > (-1)^{n-1} x_n > 0, \tag{16}$$

$$f[x_1, \ldots, x_n] \geq 0, \tag{17}$$

$k = 1, \ldots, n$,

$$f[x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n] - \left(\sum_{j \neq k} x_j \right) f[x_1, \ldots, x_n] \geq 0. \tag{18}$$

. We choose an n-tuple $(x_1, \ldots, x_n)$ that satisfies (16). By Result 20, there is a symmetric antibidiagonal matrix of the form (10), with all a_j's positive, whose spectrum is $(x_1, \ldots, x_n)$. Let us express $f(A)$ using the formula (9), with the substitutions $r_j = x_j$, $j = 1, \ldots, n$. Then, in view of Lemma 21, the $(1, \lfloor n/2 \rfloor + 1)$ entry of $f(A)$ is $a_n a_{n-1} \ldots a_2 f[x_1, \ldots, x_n]$. Since f preserves nonnegativity, and all the a_j's are positive, $f[x_1, \ldots, x_n]$ has to be nonnegative. This establishes (17).

To demonstrate (18), we look at the entries of $f(A)$ that are to the $(1, \lfloor n/2 \rfloor + 1)$ entry that was considered above. Let us fix a $k = 1, \ldots, n$. This time, however, in using formula (9) to express $f(A)$, we make the following substitutions:

$$r_j = \begin{cases} x_j & \text{if } j < k, \\ x_{j+1} & \text{if } k \leq j < n, \\ x_k & \text{if } j = n. \end{cases}$$

Our analysis splits into two cases.

Case 1. n odd. In this case, let us look at the $(1, \lfloor n/2 \rfloor + 2)$ entry of $f(A)$. By Lemma 21, and the fact that n is odd, the only power of A that contributes to this entry is A^{n-2}. Consequently

$$f(A)_{1,\lfloor n/2 \rfloor+2} = \{f[x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n] - \left(\sum_{j \neq k} x_j\right) f[x_1, \ldots, x_n]\} A^{n-2}_{1,\lfloor n/2 \rfloor+2}$$

$$= a_n a_{n-1} \ldots a_3 \{f[x_1, \ldots, x_{k-1}, x_{k+1} \ldots, x_n] - \left(\sum_{j \neq k} x_j\right) f[x_1, \ldots, x_n]\}.$$

Since f preserves nonnegativity, (18) follows from the above equalities.

Case 2. n even. In this case, we focus on the $(1, \lfloor n/2 \rfloor)$ entry of $f(A)$. We recover (18) by arguing exactly as above.

In either case, (18) is established, which concludes our proof. □

We conclude this section by showing that a subset of the necessary conditions derived above are in fact sufficient to characterize those entire functions that preserve nonnegativity of 2×2 symmetric matrices. Specifically, we show that

$$f[x_1, x_2] \geq 0 \quad \text{and}$$

$$f(x_2) - x_2 f[x_1, x_2] \geq 0 \quad \forall x_1 > -x_2 > 0$$

imply the conditions (6) and (8). This is achieved simply by taking some $y > x > 0$, making the substitutions $x_1 = y + x$ and $x_2 = x - y$, and then invoking continuity to obtain (6) and (8) for all $y \geq x \geq 0$.

9. Open problems and further ideas. We conclude this paper by listing some ideas that we did not pursue but that may lead to further progress.

One can consider matrices that preserve nonnegativity of other classes of structured matrices, such as Toeplitz or Hankel. However, since these classes are not invariant under the action of an arbitrary matrix function, their matrix functions can be quite difficult to analyze. Also, the eigenstructure of some structured matrices is rather involved, which could be an additional obstacle.

Theorem 1.3 of [35] gives an interesting formula for $f(A)$ when f is a polynomial, which therefore must also be true for entire functions. Precisely, if A is a matrix with minimal polynomial p_0 and C is the companion matrix of p_0, then

$$f(A) = \sum_{j=1}^{n} f(C)_{j1} A^{j-1}.$$

In particular, $f(A)$ is nonnegative whenever the first column of $f(C)$ is nonnegative. It would be worthwhile to find out what functions have this property.

Note that the set $\mathcal{F}_n$ contains positive constants and is closed under addition, multiplication, and composition. We are not aware of any work on systems of entire functions (or even polynomials) that satisfy this property. Perhaps one could describe a minimal set of generators (with respect to these three operations) that generate such a system.

For example, in the case $n = 1$, the generators are positive constants, the function $p_1(x) = x$, plus all quadrics of the form $(x - a)^2$, $a > 0$. Incidentally, the set

of polynomials with nonnegative coefficients is generated by positive constants and $p_1(x) = x$. We do not have a characterization of generators for $n \geq 2$.

In particular, $\mathcal{F}_n$ is a semigroup with respect to any of these operations, so some general results on semigroups may prove to be useful in our setting. Also note that the set of nonnegative matrices of order n, on which $\mathcal{F}_n$ acts, is also a semigroup (closed under addition and multiplication), which could also be of potential use.

Finally, both $\mathcal{F}_n$ and the set of nonnegative matrices of order n are also cones, so the problem might also have a cone theoretic form. If we consider polynomials instead of entire functions, we can further restrict ourselves to polynomials of degree bounded by a fixed positive integer. Then, we will obtain a proper cone, whose extreme directions may be of interest. The general problem then can also be looked upon in an appropriate similar setting.

Acknowledgments. We are grateful to Raphael Loewy, Michael Neumann, and Shmuel Friedland for helpful discussions and to anonymous referees for useful suggestions.

REFERENCES

[1] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.
[2] M. Boyle and D. Handelman, *The spectra of nonnegative matrices via symbolic dynamics*, Ann. of Math. (2), 133 (1991), pp. 249–316.
[3] M. T. Chu, *Inverse eigenvalue problems*, SIAM Rev., 40 (1998), pp. 1–39.
[4] M. T. Chu and S.-F. Xu, *On computing minimal realizable spectral radii of non-negative matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 77–86.
[5] J. B. Conway, *Functions of One Complex Variable* II, Grad. Texts in Math. 159, Springer-Verlag, New York, 1995.
[6] J. P. D'Angelo, *Inequalities from Complex Analysis*, Carus Math. Monogr. 28, Mathematical Association of America, Washington, DC, 2002.
[7] P. J. Davis, *Circulant Matrices*, John Wiley & Sons, New York, 1979.
[8] C. de Boor, *Divided differences*, Surv. Approx. Theory, 1 (2005), pp. 46–69.
[9] P. D. Egleston, T. D. Lenker, and S. K. Narayan, *The nonnegative inverse eigenvalue problem*, Linear Algebra Appl., 379 (2004), pp. 475–490.
[10] S. Friedland, *On an inverse problem for nonnegative and eventually nonnegative matrices*, Israel J. Math., 29 (1978), pp. 43–60.
[11] G. F. Frobenius, *Über Matrizen aus nicht negativen Elementen*, in Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften, Springer-Verlag, Berlin, 1912, pp. 456–477.
[12] O. Holtz, *The inverse eigenvalue problem for symmetric anti-bidiagonal matrices*, Linear Algebra Appl., 408 (2005), pp. 268–274.
[13] O. Holtz, *M-matrices satisfy Newton's inequalities*, Proc. Amer. Math. Soc., 133 (2005), pp. 711–717.
[14] O. Holtz and H. Schneider, *Open problems on GKK τ-matrices*, Linear Algebra Appl., 345 (2002), pp. 263–267.
[15] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1994.
[16] N. Jacobson, *Lectures in Abstract Algebra. Vol.* III: *Theory of Fields and Galois Theory*, D. Van Nostrand, Princeton, NJ, 1964.
[17] C. R. Johnson, *Row stochastic matrices similar to doubly stochastic matrices*, Linear and Multilinear Algebra, 10 (1981), pp. 113–130.
[18] T. J. Laffey, *Inverse eigenvalue problems for matrices*, Proc. Roy. Irish Acad. Sect. A, 95 (1995), pp. 81–88.
[19] T. J. Laffey, *Realizing matrices in the nonnegative inverse eigenvalue problem*, in Matrices and Group Representations (Coimbra, 1998), Textos Mat. Sér. B 19, Univ. Coimbra, Coimbra, 1999, pp. 21–32.
[20] T. J. Laffey and H. Šmigoc, *Nonnegative realization of spectra having negative real parts*, Linear Algebra Appl., 416 (2006), pp. 148–159.

[21] A. Leal-Duarte and C. R. Johnson, *Resolution of the symmetric nonnegative inverse eigenvalue problem for matrices subordinate to a bipartite graph*, Positivity, 8 (2004), pp. 209–213.
[22] R. Loewy and D. London, *A note on an inverse problem for nonnegative matrices*, Linear and Multilinear Algebra, 6 (1978/79), pp. 83–90.
[23] J. J. McDonald and M. Neumann, *The Soules approach to the inverse eigenvalue problem for nonnegative symmetric matrices of order $n \leq 5$*, in Algebra and Its Applications (Athens, OH, 1999), Contemp. Math. 259, AMS, Providence, RI, 2000, pp. 387–407.
[24] C. A. Micchelli and R. A. Willoughby, *On functions which preserve the class of Stieltjes matrices*, Linear Algebra Appl., 23 (1979), pp. 141–156.
[25] H. Minc, *Nonnegative Matrices*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, 1988.
[26] R. Orsi, *Numerical methods for solving inverse eigenvalue problems for nonnegative matrices*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 190–212.
[27] O. Perron, *Zur Theorie der Matrices*, Math. Ann., 64 (1907), pp. 248–263.
[28] B. A. Schmitt, *Eine explizite Darstellung der Funktion einer Dreiecksmatrix*, Z. Angew. Math. Mech., 59 (1979), pp. T76–T77.
[29] A. Seidenberg, *A new decision method for elementary algebra*, Ann. of Math. (2), 60 (1954), pp. 365–374.
[30] H. Šmigoc, *The inverse eigenvalue problem for nonnegative matrices*, Linear Algebra Appl., 393 (2004), pp. 365–374.
[31] H. Šmigoc, *Construction of nonnegative matrices and the inverse eigenvalue problem*, Linear and Multilinear Algebra, 53 (2005), pp. 85–96.
[32] R. Soto, A. Borobia, and J. Moro, *On the comparison of some realizability criteria for the real nonnegative inverse eigenvalue problem*, Linear Algebra Appl., 396 (2005), pp. 223–241.
[33] R. L. Soto, *Existence and construction of nonnegative matrices with prescribed spectrum*, Linear Algebra Appl., 369 (2003), pp. 169–184.
[34] G. W. Soules, *Constructing symmetric nonnegative matrices*, Linear and Multilinear Algebra, 13 (1983), pp. 241–251.
[35] J. D. Stafney, *Functions of a matrix and their norms*, Linear Algebra Appl., 20 (1978), pp. 87–94.
[36] J. D. Stafney, *Correction to: "Functions of a matrix and their norms"* [Linear Algebra Appl., 20 (1978), pp. 87–94], Linear Algebra Appl., 39 (1981), pp. 259–260.
[37] H. R. Suleĭmanova, *Stochastic matrices with real characteristic numbers*, Doklady Akad. Nauk SSSR (N.S.), 66 (1949), pp. 343–345 (in Russian).
[38] A. Tarski, *A Decision Method for Elementary Algebra and Geometry*, 2nd ed., University of California Press, Berkeley, Los Angeles, CA, 1951.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 102–120

THE ARNOLDI PROCESS AND GMRES FOR NEARLY SYMMETRIC MATRICES*

BERNHARD BECKERMANN† AND LOTHAR REICHEL‡

Abstract. Matrices with a skew-symmetric part of low rank arise in many applications, including path following methods and integral equations. This paper explores the properties of the Arnoldi process when applied to such a matrix. We show that an orthogonal Krylov subspace basis can be generated with short recursion formulas and that the Hessenberg matrix generated by the Arnoldi process has a structure, which makes it possible to derive a progressive GMRES method. Eigenvalue computation is also considered.

Key words. low-rank perturbation, iterative method, solution of linear system, eigenvalue computation

AMS subject classifications. 65F10, 65F15

DOI. 10.1137/060668274

1. Introduction.

This paper discusses the Arnoldi process applied to a large matrix $A \in \mathbb{R}^{n\times n}$ with a skew-symmetric part

$$A - A^* = \sum_{k=1}^{s} f_k g_k^*, \qquad f_k, g_k \in \mathbb{R}^n, \tag{1.1}$$

of low rank s. In particular, we assume that $s \ll n$. The superscript $*$ denotes transposition and, when applicable, complex conjugation. We present our results for matrices A and vectors f_k and g_k with real entries; however, our algorithms also can be applied to matrices and vectors with complex entries.

Linear systems of equations

$$Ax = b \tag{1.2}$$

with large matrices of this kind arise in path following methods, from integral equations as well as from certain boundary value problems for partial differential equations.

The generalized minimal residual (GMRES) method is one of the most popular iterative methods for the solution of large linear systems of equations with a nonsymmetric matrix. The standard implementation of GMRES is based on the Arnoldi process; see, e.g., Saad [15, section 6.5]. Application of j steps of the Arnoldi process to the matrix A with initial vector $r_0 \neq 0$ yields the decomposition

$$AV_j = V_j H_j + h_j e_j^*, \tag{1.3}$$

where $V_j = [v_1, v_2, \ldots, v_j] \in \mathbb{R}^{n\times j}$ and $h_j \in \mathbb{R}^n$ satisfy $V_j^* V_j = I_j$, $V_j^* h_j = 0$, and $v_1 = r_0/\|r_0\|$. Moreover, $H_j \in \mathbb{R}^{j\times j}$ is an upper Hessenberg matrix. Throughout

*Received by the editors August 24, 2006; accepted for publication (in revised form) by M. Benzi July 2, 2007; published electronically February 6, 2008.
http://www.siam.org/journals/simax/30-1/66827.html

†Laboratoire Painlevé UMR 8524 (ANO-EDP), UFR Mathématiques – M3, UST Lille, F-59655 Villeneuve d'Ascq CEDEX, France (bbecker@math.univ-lille1.fr). This author's research was supported in part by INTAS research network NeCCA 03-51-6637.

‡Department of Mathematical Sciences, Kent State University, Kent, OH 44242 (reichel@math.kent.edu). This author's research was supported in part by NSF grant DMS-0107858 and an OBR Research Challenge Grant.

this paper I_j denotes the identity matrix of order j, e_k denotes the kth column of an identity matrix of appropriate order, and $\|\cdot\|$ denotes the Euclidean vector norm. For ease of discussion, we will assume that j is small enough so that the decomposition (1.3) with the stated properties exists.

When $h_j \neq 0$, we can express (1.3) in the form

$$AV_j = V_{j+1}\bar{H}_j, \tag{1.4}$$

where $v_{j+1} = h_j/\|h_j\|$ and

$$V_{j+1} = [V_j, v_{j+1}] \in \mathbb{R}^{n\times(j+1)}, \qquad \bar{H}_j = \begin{bmatrix} H_j \\ \|h_j\|e_j^* \end{bmatrix} \in \mathbb{R}^{(j+1)\times j}.$$

The computation of the Arnoldi decompositions (1.3) or (1.4) of a general $n \times n$ matrix A requires the evaluation of j matrix-vector products with A and of about $j^2/2$ inner products with n-vectors. The latter demands $\mathcal{O}(nj^2)$ arithmetic floating point operations (flops) and may dominate the computational work. The Arnoldi process determines the columns of V_j in order and requires access to all the previously generated columns to compute the next one; in particular, all the columns of V_j have to be stored; see, e.g., Saad [15, section 6.3] for a thorough treatment of the Arnoldi process. Computation of the jth iterate by GMRES also requires the whole matrix V_j to be available. To limit the demand of computer memory, GMRES is often restarted periodically, say, every m steps. This restarted GMRES method is denoted by GMRES(m). Restarting may reduce the rate of convergence of GMRES significantly.

In section 2, we show that the property (1.1) of A makes it possible to determine the columns v_k of V_j with a short recursion formula, the number of terms of which depends on s in (1.1) but can be bounded independently of k. The recursion formula allows the computation of all the columns of V_j in only $\mathcal{O}(nj)$ flops. Moreover, the computation of v_k for large k does not require access to all the previously computed columns of V_j. Section 3 discusses the structure of the Hessenberg matrix H_j in (1.3) when A satisfies (1.1) and presents a fast algorithm for determining the Arnoldi decomposition (1.4).

The short recursion formula for the columns of V_j and the structure of H_j make it possible to derive a progressive GMRES method for the solution of linear systems (1.2) with a matrix that satisfies (1.1). Such a method is described in section 4. The storage requirement of the method, as well as the computational effort per iteration, are bounded independently of the number of iterations j. This makes it possible to apply the method without periodic restarts. Computed examples are presented in section 5 and concluding remarks can be found in section 6.

Recently, Barth and Manteuffel [4] presented iterative methods of conjugate gradient type for linear systems of equations of the kind considered in the present paper. Specifically, they considered linear systems of equations with a generalized B-normal(ℓ, m) matrix. This type of matrix is characterized by the existence of polynomials p_ℓ and q_m of degrees ℓ and m, respectively, such that the matrix

$$A^\dagger q_m(A) - p_\ell(A)$$

is of low rank, where $A^\dagger = B^{-1}A^*B$ and B is a Hermitian positive definite matrix. The matrix $A^\dagger$ is the adjoint of A with respect to the B-inner product

$$\langle u, v\rangle_B = u^*Bv.$$

In the terminology of Barth and Manteuffel [4] matrices A that satisfy (1.1) are generalized I-normal$(1,0)$ matrices.

Barth and Manteuffel [4] derived their methods by generalizing the recurrence relations for orthogonal polynomials on the unit circle. The latter type of recurrence relations had previously been applied to iterative methods in [11, 12]; see also Arnold et al. [2] for a recent application to QCD computations. The derivation of our iterative methods for (1.2) differs from the derivation by Barth and Manteuffel [4] of their schemes in that we do not apply properties of orthogonal polynomials on the unit circle. Iterative methods for linear systems of equations with a matrix, whose symmetric part is positive definite and easily invertible, are described by Concus and Golub [7] and Widlund [18].

2. Generation of an orthogonal Krylov subspace basis. Introduce the Krylov subspace

$$\mathcal{K}_j(A,b) = \text{span}\{b, Ab, A^2b, \ldots, A^{j-1}b\}, \tag{2.1}$$

which we assume to be of dimension j. The columns of the matrix V_j in (1.3) form an orthonormal basis of $\mathcal{K}_j(A,b)$.

Let f_k and g_k be the vectors in (1.1) and define the matrices

$$F = [f_1, f_2, \ldots, f_s], \qquad G = [g_1, g_2, \ldots, g_s], \tag{2.2}$$

which we may assume to be of full rank; otherwise we can reduce s. We express (1.1) as

$$A - A^* = FG^* \tag{2.3}$$

and note that

$$FG^* = -GF^*. \tag{2.4}$$

It follows from (2.4) and the fact that F and G are of full rank that s is even and that there is a unique matrix $C \in \mathbb{R}^{s\times s}$, such that

$$G = FC. \tag{2.5}$$

The fact that s is even can be seen by substituting (2.5) into (2.4). This yields $C^* = -C$. Therefore, when s is odd, C is singular and G is not of full rank. Use of the representation (2.5) of G reduces the computational work in the algorithms presented in sections 3 and 4.

2.1. In many applications that involve a matrix A with a skew-symmetric part of low rank, the matrix is given in the form

$$A = M + \sum_{k=1}^{s/2} f_k g_k^*$$

with $M \in \mathbb{R}^{n\times n}$ symmetric. Then (1.1) can be expressed as

$$A - A^* = \sum_{k=1}^{s/2} f_k g_k^* - \sum_{k=1}^{s/2} g_k f_k^*$$

and we may choose

$$F = [f_1, f_2, \ldots, f_{s/2}, g_1, g_2, \ldots, g_{s/2}], \qquad C = \begin{bmatrix} 0 & -I_{s/2} \\ I_{s/2} & 0 \end{bmatrix}.$$

Introduce the vectors

$$f_{\ell,k} = V_k V_k^* f_\ell, \qquad 1 \le \ell \le s, \qquad 1 \le k \le j. \tag{2.6}$$

Then

$$f_{\ell,k} \in \mathcal{K}_k(A, r_0), \qquad f_{\ell,k} - f_\ell \perp \mathcal{K}_k(A, r_0). \tag{2.7}$$

Moreover, for each ℓ, the $f_{\ell,k}$ satisfy the recursion

$$\begin{cases} f_{\ell,k} = f_{\ell,k-1} + v_k v_k^* f_\ell, & k = 2, 3, \ldots, j, \\ f_{\ell,1} = v_1 v_1^* f_\ell. \end{cases} \tag{2.8}$$

Let

$$v_k' = A^* v_k + \sum_{\ell=1}^{s} g_\ell^* v_k (f_\ell - f_{\ell,k}). \tag{2.9}$$

Then (1.1) gives

$$v_k' - A v_k = \sum_{\ell=1}^{s} g_\ell^* v_k (f_\ell - f_{\ell,k}) - (A - A^*) v_k = -\sum_{\ell=1}^{s} g_\ell^* v_k f_{\ell,k}. \tag{2.10}$$

We may assume that $Av_k \notin \mathcal{K}_k(A, r_0)$, because otherwise range(V_k) is an invariant subspace of A, which contains the solution of the linear system (1.2); see, e.g., Saad [15, section 6.5.4] for details. The following properties of v_k' are a consequence of the above discussion.

PROPOSITION 2.2. [illegible] v_k' [illegible] (2.9) [illegible] [illegible] $\dim \mathcal{K}_{k+1}(A, r_0) = k + 1$ [illegible]

$$v_k' \in \mathcal{K}_{k+1}(A, r_0) \setminus \mathcal{K}_k(A, r_0), \qquad v_k' \perp \mathcal{K}_{k-2}(A, r_0). \tag{2.11}$$

[illegible]. The requirement that $\mathcal{K}_{k+1}(A, r_0)$ be of dimension $k + 1$ secures that $Av_k \notin \mathcal{K}_k(A, r_0)$. Equation (2.7) yields that $v_k' - Av_k \in \mathcal{K}_k(A, r_0)$, and this establishes the left-hand side of (2.11).

It follows from the Arnoldi decomposition (1.3) that $v_k \perp Av_\ell$ for $1 \le \ell \le k-2$, or, equivalently, that $A^* v_k \perp v_\ell$ for $1 \le \ell \le k-2$. The latter property, in combination with (2.7) and (2.9), shows the orthogonality relation (2.11). □

Equation (2.10) yields the expression

$$v_k' = A v_k - \sum_{\ell=1}^{s} g_\ell^* v_k f_{\ell,k}, \tag{2.12}$$

which we use to evaluate v_k'. Orthogonalization against the vectors v_{k-1} and v_k, and normalization of the resulting vector, gives the Arnoldi vector v_{k+1}. In what follows we will write this operation more explicitly as

$$v_k' = t_{k+1,k} v_{k+1} + t_{k,k} v_k + t_{k-1,k} v_{k-1}, \qquad k \ge 1, \tag{2.13}$$

where

$$t_{k-1,k} = v_{k-1}^* v_k', \quad t_{k,k} = v_k^* v_k', \quad t_{k+1,k} = v_{k+1}^* v_k', \tag{2.14}$$

with $v_0 = 0$ and $t_{k+1,k} = \|v_k' - t_{k,k} v_k - t_{k,k-1} v_{k-1}\| > 0$. The computations for generating the orthogonal Krylov subspace basis, and for determining the matrix $\bar{H}_j$ in (1.4), are summarized in Algorithm 3.2 of the following section.

3. Structure of the Hessenberg matrices. This section discusses the structure of the matrices $H_j = [h_{k,\ell}]$ and $\bar{H}_j = [h_{k,\ell}]$ in the Arnoldi decompositions (1.3) and (1.4), respectively. It is convenient to introduce the following terminology. For an integer m, the m-diagonal of a matrix $B = [b_{k,\ell}]$ consists of all entries of the form $b_{k,k+m}$. The m-upper (m-lower) triangular part of B is the submatrix comprising all entries on and above (below) the m-diagonal. For instance, the upper Hessenberg matrices H_j and $\bar{H}_j$ have vanishing (-2)-lower triangular parts. Note that the (-2)-upper triangular part is not triangular.

PROPOSITION 3.1. [illegible] $\hat{F}_j = V_j^* F$ [illegible] $\hat{G}_j = V_j^* G$ [illegible] F [illegible] G [illegible] (2.3) [illegible] $j \geq s$ [illegible] H_j [illegible] (1.3) [illegible]

$$H_j - H_j^* = \hat{F}_j \hat{G}_j^*, \qquad \hat{G}_j \hat{F}_j^* = -\hat{F}_j \hat{G}_j^*, \tag{3.1}$$

[illegible] H_j [illegible] s [illegible] H_j [illegible] $\hat{F}_j \hat{G}_j^*$ [illegible] 2 [illegible]. It follows from (1.3) and (2.3) that

$$H_j = V_j^* A V_j = V_j^*(A^* + FG^*)V_j = H_j^* + \hat{F}_j \hat{G}_j^*,$$

which shows (3.1). Since the (-2)-lower triangular part of H_j vanishes, (3.1) yields the 2-upper triangular part of H_j. □

The proposition shows that H_j is an order-$(1, s+1)$ quasi-separable matrix; see, e.g., Eidelman, Gohberg, and Olshevsky [9] for a recent discussion on this kind of matrix.

We turn to the entries in the tridiagonal part of $\bar{H}_j$. In accordance with (2.14), we define the matrix $\bar{T}_j = [t_{m,k}] \in \mathbb{R}^{(j+1)\times j}$ with entries $t_{m,k} = v_m^* v_k'$. Notice that $\bar{T}_j$ is tridiagonal by Proposition 2.2. Substitution of (2.6) into (2.12) gives

$$v_k' = Av_k - \sum_{\ell=1}^{s} (g_\ell^* v_k) V_k V_k^* f_\ell = Av_k - V_k V_k^* F G^* v_k = Av_k - V_k \hat{F}_k \hat{G}_k^* e_k,$$

and, taking into account that $e_m^* \hat{F}_k \hat{G}_k^* e_k = e_m^* \hat{F}_j \hat{G}_j^* e_k$ for $m \leq k \leq j$, we get for the entries $h_{m,k} = v_m^* A v_k$ of $\bar{H}_j$ the formula

$$h_{m,k} = \begin{cases} t_{k+1,k}, & m = k+1, \\ t_{m,k} + e_m^* \hat{F}_j \hat{G}_j^* e_k, & k-1 \leq m \leq k, \\ e_m^* \hat{F}_j \hat{G}_j^* e_k, & 1 \leq m < k-1. \end{cases} \tag{3.2}$$

Thus, the matrix $\hat{F}_j \hat{G}_j^*$ contributes to the upper triangular part of $\bar{H}_j$, and the matrix $\bar{T}_j$, which expresses the orthogonalization of the vectors v_k', contributes to the tridiagonal part; in MATLAB notation, we have

$$\bar{H}_j = \bar{T}_j + \mathsf{triu}(\hat{F}_j \hat{G}_j^*, 0).$$

Combining (2.9) with (2.7) yields

$$v_m^* v_k' = v_m^* A^* v_k = (v_k^* A v_m)^*, \qquad 1 \leq m \leq k,$$

and comparison with (2.14) gives

$$t_{k-1,k} = t_{k,k-1} > 0, \qquad t_{k,k} = h_{k,k}^*. \tag{3.3}$$

We describe an algorithm for the computation of the matrices $\bar{H}_j$ and V_{j+1} in the decomposition (1.4), assuming that the decomposition exists. The matrix $\bar{H}_j$ is represented in decomposed form (3.2) by the matrices $\hat{F}_j$, $\hat{G}_j$, and $\bar{T}_j$, which in the algorithm are represented without subscript j. The subscripts used in the algorithm denote row and column indices. Thus, $\hat{F}_{k,:}$ denotes the kth row of the matrix $\hat{F}_j$. At iteration k, we let $\tilde{F} = [f_{1,k}, f_{2,k}, \ldots, f_{s,k}]$.

ALGORITHM 3.2. [illegible] $\bar{H}_j$ [illegible] V_{j+1}

[illegible] $A \in \mathbb{R}^{n\times n}$, $F = [f_1, f_2, \ldots, f_s], G = [g_1, g_2, \ldots, g_s] \in \mathbb{R}^{n\times s}$, $r_0 \in \mathbb{R}^n$, j

[illegible] $\bar{T} = [t_{\ell,k}] \in \mathbb{R}^{(j+1)\times j}$, $\hat{F}, \hat{G} \in \mathbb{R}^{(j+1)\times s}$, $V_{j+1} = [v_1, v_2, \ldots, v_{j+1}] \in \mathbb{R}^{n\times(j+1)}$

1. $\tilde{F} := 0$
2. $v_1 := r_0/\|r_0\|$
3. [illegible] $k = 1 : j$
   4. $\hat{F}_{k,:} := v_k^* F$, $\hat{G}_{k,:} := v_k^* G$
   5. $\tilde{F} := \tilde{F} + v_k \hat{F}_{k,:}$
   6. $v' := Av_k - \tilde{F}\hat{G}_{k,:}^*$
   7. [illegible] $k > 1$ [illegible]
      8. $t_{k-1,k} := v_{k-1}^* v'$, $v' := v' - t_{k-1,k} v_{k-1}$
   9. [illegible]
   10. $t_{k,k} := v_k^* v'$, $v' := v' - t_{k,k} v_k$, $t_{k+1,k} := \|v'\|$, $v_{k+1} := v'/t_{k+1,k}$
11. [illegible]

We note that the computational effort of line 4 of the algorithm can be essentially halved by using the representation (2.5) of G.

Algorithm 3.2 can be applied to compute approximations of a few extreme eigenvalues and associated eigenvectors of A similarly to the standard implementation of the Arnoldi process. Certain eigenvalues of H_j are used to approximate selected eigenvalues of A. The structure of H_j therefore is of interest.

[illegible] 3.3. Given a unitary matrix $Q \in \mathbb{C}^{j\times j}$, it follows from Proposition 3.1 that for the matrix

$$S = Q^* H_j Q,$$

we have $\Sigma := S - S^* = Q^* \hat{F}_j \hat{G}_j^* Q$, i.e., S has a skew-symmetric part of rank s. If S has an additional sparsity structure, then we may derive results similarly to Proposition 3.1. For instance, the matrix S in the Schur normal form of H_j is upper triangular, and thus S may be written as a diagonal matrix plus the 1-upper triangular part of the matrix Σ. Similarly, the matrix S obtained after one step of the QR-algorithm is upper Hessenberg and therefore may be written as a tridiagonal matrix plus the 2-upper triangular part of the matrix Σ.

We recall that in the QR-algorithm for eigenvalue computations the unitary factor Q is chosen such that $R = Q^* H_j$ is upper triangular.

[illegible] 3.4. Consider the QR-decomposition $H_j = QR$ with orthogonal Q and upper triangular R. Here also the matrix R has a structure: since Q^* is known to be of lower Hessenberg form (see, e.g., the considerations of the next section), we see from Proposition 3.1 that the 3-upper triangular part of $Q^*(H_j - \hat{F}_j \hat{G}_j)$ contains only zeros, or, in other words, the 3-upper triangular parts of R_j and of the matrix $Q^* \hat{F}_j \hat{G}_j$ of rank s coincide.

The structure makes it possible to compute the matrix R in $\mathcal{O}(j)$ flops, by representing H_j in terms of the tridiagonal part of H_j and the matrices $\hat{F}_j$ and $\hat{G}_j$, and by representing R in terms of its 0-, 1-, and 2-diagonals and the matrices $Q^* \hat{F}_j$ and $\hat{G}_j$.

Since the computation of R does not play a role in subsequent considerations, we omit the details.

4. A progressive GMRES algorithm. Let $x_0 \in \mathbb{R}^n$ be an approximate solution of (1.2). GMRES determines a new approximate solution x_j of (1.2), such that

$$\|Ax_j - b\| = \min_{x \in x_0 + \mathcal{K}_j(A,r_0)} \|Ax - b\|, \qquad x_j \in x_0 + \mathcal{K}_j(A, r_0). \tag{4.1}$$

The standard implementation of GMRES determines a correction of x_0, i.e., $x_j = x_0 + V_j x_j$, by substituting the decomposition (1.4) with $r_0 = b - Ax_0$ into (4.1); see, e.g., Saad [15, section 6.5] for details. This gives the equivalent minimization problem

$$\min_{y \in \mathbb{R}^j} \|\bar{H}_j y - e_1 \|r_0\| \|, \tag{4.2}$$

with solution $y_j \in \mathbb{R}^j$.

We solve the least-squares problem (4.2) by using the QR-factorization $\bar{H}_j = Q_{j+1}\bar{R}_j$, where $Q_{j+1} \in \mathbb{R}^{(j+1)\times(j+1)}$ is orthogonal (or unitary in the case of complex A, b) and

$$\bar{R}_j = \begin{bmatrix} R_j \\ 0 \end{bmatrix} \in \mathbb{R}^{(j+1)\times j}, \tag{4.3}$$

with $R_j \in \mathbb{R}^{j\times j}$ upper triangular. Let us first recall in the following paragraph and Proposition 4.1 the well-known construction of a QR-decomposition of the upper Hessenberg matrix $\bar{H}_j$ for a general matrix A. Subsequently, we explain in Proposition 4.2 how the structure of the matrix A helps us to derive a progressive form of GMRES.

Following Saad [15, Chapter 6.5.3], we determine the matrix Q_{j+1} by applying a product of Givens rotations to $\bar{H}_j$. Let $Q_1 = [1]$ and define, for $k = 1, 2, \ldots, j$,

$$Q_{k+1}^* = \Omega_{k+1} \begin{bmatrix} Q_k^* & 0 \\ 0 & 1 \end{bmatrix}, \qquad \Omega_{k+1} = \begin{bmatrix} I_{k-1} & 0 & 0 \\ 0 & c_k^* & s_k \\ 0 & -s_k & c_k \end{bmatrix}, \tag{4.4}$$

with $s_k \geq 0$ and $s_k^2 + |c_k|^2 = 1$ such that Ω_{k+1} is unitary (and reduces to a classical Givens rotation in the case of real data). Using the nested structure of $\bar{H}_j = [h_{k,\ell}]$, i.e., the fact that $\bar{H}_{j-1}$ is the leading $j \times (j-1)$ principal submatrix of $\bar{H}_j$, yields

$$Q_{j+1}^* \bar{H}_j = \Omega_{j+1} \begin{bmatrix} Q_j^* \bar{H}_{j-1} & Q_j^* H_j e_j \\ 0 & h_{j+1,j} \end{bmatrix} = \Omega_{j+1} \begin{bmatrix} R_{j-1} & * \\ 0 & \tau_j \\ 0 & h_{j+1,j} \end{bmatrix},$$

with

$$\tau_j = e_j^* Q_j^* H_j e_j. \tag{4.5}$$

Since multiplication by Ω_{j+1} affects only the last two rows, the matrices R_j and $\bar{R}_j$ also have a nested substructure:

$$\bar{R}_j = \begin{bmatrix} \bar{R}_{j-1} & * \\ 0 & 0 \end{bmatrix}, \qquad R_j = \begin{bmatrix} R_{j-1} & * \\ 0 & * \end{bmatrix}.$$

We have the following formulas for the coefficients c_j, s_j of Ω_{j+1} and for the entries of Q^*_{j+1}.

PROPOSITION 4.1.

$$s_j = \frac{t_{j+1,j}}{\sqrt{t_{j+1,j}^2 + |\tau_j|^2}} \geq 0, \qquad c_j = \frac{\tau_j}{\sqrt{t_{j+1,j}^2 + |\tau_j|^2}}, \tag{4.6}$$

$t_{j+1,j} = h_{j+1,j}$ $\bar{H}_j$ τ_j (4.5). j Q^*_{j+1} Q^*_j Q^*_{j+1} $j \geq 3$

$$e^*_{j+1} Q^*_{j+1} = [\, -s_j e^*_j Q^*_j,\ c_j \,] = [\, *\,,\ s_j s_{j-1} c_{j-2}\,,\ -s_j c_{j-1}\,,\ c_j \,]. \tag{4.7}$$

The proof is obtained by direct calculations. □

We are in a position to describe a progressive recurrence relation for the GMRES residual r_j, a simplified recurrence for its norm, as well as a simplified expression for the quantity τ_j defined by (4.5). In particular, the progressive GMRES algorithm does not require the entries of the matrices R_j, $\bar{H}_j$, and Q_{j+1}. Only the c_k, s_k of the Givens rotations (4.4) and the quantities occurring in the recurrence relation for the Arnoldi vectors v_k are needed.

PROPOSITION 4.2. r_j x_j

$$r_j = b - Ax_j, \tag{4.8}$$

$$\gamma_j = -s_j \gamma_{j-1}, \qquad j \geq 1, \tag{4.9}$$

$\gamma_0 = \|r_0\|$ $\gamma_j = (-1)^j \|r_j\|$

$$r_j = s_j^2 r_{j-1} + \gamma_j c_j^* v_{j+1}, \qquad j \geq 1. \tag{4.10}$$

$p_j \in \mathbb{R}^s$

$$p_j^* = -s_{j-1} p_{j-1}^* + c_{j-1} e_j^* \hat{F}_j, \qquad j \geq 2, \tag{4.11}$$

$p_1^* = \hat{F}_1$ τ_j (4.5)

$$\tau_j = c_{j-1} t_{j,j} - s_{j-1} c_{j-2} t_{j-1,j} + p_j^* \hat{G}_j^* e_j, \qquad j \geq 2, \tag{4.12}$$

$c_0 = 1$ $\tau_1 = t^*_{1,1}$

We start by establishing the formula

$$r_j = \gamma_j V_{j+1} Q_{j+1} e_{j+1}. \tag{4.13}$$

A different proof is presented by Saad [15, Proposition 6.9]. From the definition of GMRES, we have that $r_j = P^{\perp}_{A\mathcal{K}_j(A,r_0)} r_0$, where $P_{A\mathcal{K}_j(A,r_0)}$ denotes the orthogonal projector onto $A\mathcal{K}_j(A, r_0)$ and $P^{\perp}_{A\mathcal{K}_j(A,r_0)} = I - P_{A\mathcal{K}_j(A,r_0)}$ denotes the orthogonal projector onto the complement. Denote by $\bar{Q}_j \in \mathbb{R}^{(j+1)\times j}$ the matrix made up of the first j columns of Q_{j+1}. From (1.4) and (4.3), we obtain that

$AV_j = V_{j+1}Q_{j+1}\bar{R}_j = V_{j+1}\bar{Q}_j R_j$. Since R_j is invertible, we see that an orthonormal basis of $A\mathcal{K}_j(A, r_0)$ is given by the columns of $V_{j+1}\bar{Q}_j$, implying that

$$\begin{aligned} r_j &= r_0 - P_{A\mathcal{K}_j(A,r_0)}r_0 = V_{j+1}Q_{j+1}Q_{j+1}^*V_{j+1}^*r_0 - V_{j+1}\bar{Q}_j\bar{Q}_j^*V_{j+1}^*r_0 \\ &= V_{j+1}(Q_{j+1}Q_{j+1}^* - \bar{Q}_j\bar{Q}_j^*)e_1\|r_0\| = V_{j+1}Q_{j+1}e_{j+1}e_{j+1}^*Q_{j+1}^*e_1\|r_0\|. \end{aligned}$$

It follows from (4.7) and (4.9) that

$$\gamma_0\, e_{j+1}^*Q_{j+1}^*e_1 = \gamma_0(-s_j)\, e_j^*Q_j^*e_1 = \cdots = \gamma_0(-s_j)(-s_{j-1})\ldots(-s_1) = \gamma_j.$$

This establishes (4.13). Since $V_{j+1}Q_{j+1}$ has orthonormal columns and $s_k \geq 0$ by Proposition 4.1, we may conclude by taking norms in (4.13) that $|\gamma_j| = \|r_j\| = (-1)^j\gamma_j$.

The updating formula (4.10) is now an immediate consequence of (4.13): by (4.7),

$$r_j = \gamma_j V_{j+1}[-s_j e_j^*Q_j^*, c_j]^* = -s_j\frac{\gamma_j}{\gamma_{j-1}}r_{j-1} + \gamma_j c_j^* v_{j+1}.$$

It remains to show (4.12). From (4.7) and (4.11) we conclude by recurrence on j that

$$p_j^* = e_j^*Q_j^*\hat{F}_j, \qquad j \geq 1.$$

The structure of H_j, together with (4.7) and (4.13), yields for $j \geq 2$ that

$$\begin{aligned} \tau_j &= e_j^*Q_j^*H_je_j \\ &= e_j^*Q_j^*\left(\begin{bmatrix} 0 \\ \vdots \\ 0 \\ t_{j-1,j} \\ t_{j,j} \end{bmatrix} + \hat{F}_j\hat{G}_j^*e_j\right) = [-s_{j-1}c_{j-2}, c_{j-1}]\begin{bmatrix} t_{j-1,j} \\ t_{j,j} \end{bmatrix} + p_j^*\hat{G}_j^*e_j. \end{aligned}$$

When $j = 1$, we get by using $Q_1 = [1]$ and (3.3) that $\tau_1 = h_{1,1} = t_{1,1}^*$. ∎

By applying a suitable linear operator L, such that $Lr_k = x_k$ for $0 \leq k \leq j+1$, to the recurrence relation (4.10) of the residuals, we obtain an updating formula for the GMRES iterates in terms of the auxiliary vectors $z_k = Lv_k$ and $w_{\ell,k} = Lf_{\ell,k}$, which together with the recursive computation of these new vectors is described in the following proposition.

PROPOSITION 4.3. [illegible] $\dim\mathcal{K}_{j+1}(A, r_0) = j+1$ [illegible]

$$w_{\ell,k} = w_{\ell,k-1} + v_k^*f_\ell z_k, \quad 0 < k \leq j, \tag{4.14}$$

$$z_{k+1} = -\frac{1}{t_{k+1,k}}\left(v_k + t_{k,k}z_k + t_{k-1,k}z_{k-1} + \sum_{\ell=1}^{s} g_\ell^*v_k w_{\ell,k}\right), \quad 1 < k \leq j, \tag{4.15}$$

[illegible]

$$w_{\ell,0} = 0, \qquad z_1 = \frac{x_0}{\gamma_0}, \qquad z_2 = -\frac{1}{t_{2,1}}(v_1 + t_{1,1}^*z_1). \tag{4.16}$$

[illegible] $0 < k \leq j$ [illegible]

$$x_k = s_k^2x_{k-1} + \gamma_k c_k^*z_{k+1}. \tag{4.17}$$

. Consider the QR-factorization

$$[r_0, Ar_0, \ldots, A^j r_0] = V_{j+1} S_{j+1},$$

i.e., $S_{j+1} \in \mathbb{R}^{(j+1)\times(j+1)}$ is upper triangular and invertible by assumption on j. The projector

$$P = V_{j+1} S_{j+1} (I_{j+1} - e_1 e_1^*) S_{j+1}^{-1} V_{j+1}^*$$

satisfies

$$P\left(\sum_{k=0}^{j} \alpha_k A^k r_0\right) = \sum_{k=1}^{j} \alpha_k A^k r_0, \qquad (I-P)\left(\sum_{k=0}^{j} \alpha_k A^k r_0\right) = \alpha_0 r_0.$$

As a consequence, defining the linear operator L by

$$Lv = \frac{r_0^*(I-P)v}{r_0^* r_0} x_0 - A^{-1} P v,$$

we get for any $u \in \mathcal{K}_j(A, r_0)$ that

$$L(b - A(x_0 + u)) = x_0 + u.$$

In particular, we obtain $Lr_k = x_k$ for $0 \le k \le j$, as claimed above. In order to see that the vectors z_{k+1} and $w_{\ell,k}$ defined by

$$z_{k+1} = Lv_{k+1}, \qquad w_{\ell,k} = Lf_{\ell,k}, \qquad 0 \le k \le j,$$

can be computed via the relations (4.14)–(4.16), we argue by recurrence on k: applying L to the relations $f_{\ell,0} = 0$, $v_1 = r_0/\gamma_0 = (b - Ax_0)/\gamma_0$, and $Av_1 = h_{2,1}v_2 + h_{1,1}v_1 = t_{2,1}v_2 + t_{1,1}^* v_1$, respectively, leads to the initializations (4.16). Similarly, for (4.14) we apply L to (2.8), and (4.15) is obtained by applying L both to (2.12) and (2.13), where we notice that $L(Av_k) = -v_k$. Finally, the recurrence relation (4.17) for the GMRES iterates follows by applying L to (4.10). ∎

Let $W_j = [w_{1,j}, w_{2,j}, \ldots, w_{s,j}] \in \mathbb{R}^{n\times s}$. Then (4.14) can be written as

$$W_j = W_{j-1} + z_j e_j^* \hat{F}_j, \qquad W_1 = \frac{x_0}{\gamma_0}\hat{F}_1,$$

and

$$\sum_{\ell=1}^{s} g_\ell^* v_j \, w_{\ell,j} = W_j \hat{G}_j^* e_j.$$

Algorithm 4.4 below works with the matrices W_j rather than with their columns individually. The notation of Algorithm 4.4 follows that of Algorithm 3.2. In particular, the matrices W_j are stored in W.

ALGORITHM 4.4.

$A \in \mathbb{R}^{n\times n}$, $F = [f_1, f_2, \ldots, f_s], G = [g_1, g_2, \ldots, g_s] \in \mathbb{R}^{n\times s}$, $b, x_0 \in \mathbb{R}^n$

$x_j \in \mathbb{R}^n$

%

1. $r_0 := b - Ax_0$ $\gamma_0 := \|r_0\|$
2. $v_1 := r_0/\gamma_0$ $z_1 := x_0/\gamma_0$

% $j = 1$
3. $\hat{F}_{1,:} := v_1^* F$, $\hat{G}_{1,:} := v_1^* G$
4. $p_1^* := \hat{F}_{1,:}$, $\tilde{F} := v_1 \hat{F}_{1,:}$, $W := x_0 \hat{F}_1 / \gamma_0$
5. $v' := Av_1 - \tilde{F}\hat{G}_{1,:}^*$
6. $t_{1,1} := v_1^* v'$, $v' := v' - t_{1,1} v_1$
7. $t_{2,1} := \|v'\|$, $v_2 := v'/t_{2,1}$
8. $\tau_1 := t_{1,1}^*$
9. $c_1 := \tau_1/(|\tau_1|^2 + t_{2,1}^2)^{1/2}$, $s_1 := t_{2,1}/(|\tau_1|^2 + t_{2,1}^2)^{1/2}$, $\gamma_1 := -s_1 \gamma_0$
10. $z_2 := -(v_1 + t_{1,1}^* z_1)/t_{2,1}$, $x_1 := s_1^2 x_0 + \gamma_1 c_1^* z_2$
% $j > 1$
11. for $j = 2, 3, \ldots$ [illegible]
12. $\hat{F}_{j,:} := v_j^* F$, $\hat{G}_{j,:} := v_j^* G$
13. $p_j^* := -s_{j-1} p_{j-1}^* + c_{j-1} \hat{F}_{j,:}$, $\tilde{F} := \tilde{F} + v_j \hat{F}_{j,:}$, $W := W + z_j \hat{F}_{j,:}$
14. $v' := Av_j - \tilde{F}\hat{G}_{j,:}^*$
15. $t_{j-1,j} := v_{j-1}^* v'$, $v' := v' - t_{j-1,j} v_{j-1}$
16. $t_{j,j} := v_j^* v'$, $v' := v' - t_{j,j} v_j$, $t_{j+1,j} := \|v'\|$, $v_{j+1} := v'/t_{j+1,j}$
17. $\tau_j := c_{j-1} t_{j,j} - s_{j-1} c_{j-2} t_{j-1,j} + p_j^* \hat{G}_{j,:}^*$
18. $c_j := \tau_j/(|\tau_j|^2 + t_{j+1,j}^2)^{1/2}$, $s_j := t_{j+1,j}/(|\tau_j|^2 + t_{j+1,j}^2)^{1/2}$, $\gamma_j := -s_j \gamma_{j-1}$
19. $z_{j+1} := -(v_j + t_{j,j} z_j + t_{j-1,j} z_{j-1} + W\hat{G}_{j,:}^*)/t_{j+1,j}$
20. $x_j := s_j^2 x_{j-1} + \gamma_j c_j^* z_{j+1}$
21. [illegible]

Iterations with GMRES are typically terminated when the residual vector (4.8) is sufficiently small, e.g., when

$$\|r_j\|/\|r_0\| \leq \varepsilon \tag{4.18}$$

for a user-specified value of ε. This stopping criterion can be easily evaluated, since Algorithm 4.4 computes γ_j, with $|\gamma_j| = \|r_j\|$, in each iteration. If the residual vectors are desired in each iteration, then one can add the relation (4.10) on line 10 (for $j = 1$) and on line 20 of the algorithm. Stopping criteria of the type (4.18) have recently been discussed by Paige et al. [13, 14]. In particular, the initial vector x_0 should be chosen so that $\|r_0\| \leq \|b\|$ and preferably as the zero-vector.

In order to make the connection between Algorithm 4.4 and the preceding discussion clearer, vectors are equipped with subscripts in the algorithm. However, only the most recently generated vectors p_j^* and x_j have to be stored simultaneously, and only the two most recently generated vectors v_j, v_{j-1} and z_j, z_{j-1} have to be stored at any given time. Only the jth rows of the matrices $\hat{F}$ and $\hat{G}$ have to be stored simultaneously. The matrices $\tilde{F}$ and W have to be stored and require $n \times s$ storage locations each. Moreover, representations of the matrices A, F, and G have to be stored. Ignoring the storage for the latter, the storage requirement for Algorithm 4.4 is bounded by $(2s+6)n + \mathcal{O}(sj)$ storage locations. The computational work per iteration is bounded independent of j; it is $\mathcal{O}(n)$ flops in addition to the arithmetic work required for the evaluation of Av_j. In the special case when $s = 0$, Algorithm 4.4 simplifies to a minimal residual method for the solution of linear systems of equations with a symmetric, possibly indefinite, matrix.

We conclude this section with a comment on FOM, an iterative method that is closely related to GMRES; see Saad [15, section 6.4]. The jth iterate determined by

FOM, $x_j^{\mathrm{FOM}} \in x_0 + \mathcal{K}_j(A, r_0)$, satisfies

$$b - A\, x_j^{\mathrm{FOM}} \perp \mathcal{K}_j(A, r_0).$$

From, e.g., [15, section 6.5.5] we know that the iterate x_j^{FOM} exists if and only if $|s_j| = \|r_{j-1}\|/\|r_j\| < 1$, which is equivalent to $c_j \neq 0$, where s_j and c_j are entries of the Givens rotation Ω_{j+1}; see (4.4). In this case, the relation between x_j^{FOM} and the GMRES iterate x_j is given by

$$x_j = s_j^2\, x_{j-1} + (1 - s_j^2)\, x_j^{\mathrm{FOM}};$$

see Saad [15, section 6.5.5] for details. A comparison with (4.10) shows that

$$x_j^{\mathrm{FOM}} = \frac{\gamma_j}{c_j} z_{j+1},$$

i.e., the vectors z_{j+1} are FOM iterates up to normalization.

5. Computed examples. Linear systems of equations (1.2) with matrices of the form

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \in \mathbb{R}^{n\times n},$$

with a symmetric leading principal submatrix $A_{1,1} \in \mathbb{R}^{(n-\ell)\times(n-\ell)}$ and $A_{1,2}, A_{2,1}^* \in \mathbb{R}^{(n-\ell)\times\ell}$, $A_{2,2} \in \mathbb{R}^{\ell\times\ell}$, arise in many applications. Example 5.1 outlines a path following method that gives rise to matrices of this kind, and Examples 5.2–5.4 discuss the solution of integral equations. All computations were carried out in MATLAB with machine epsilon about $2 \cdot 10^{-16}$.

5.1. We are interested in computing the solution u of the nonlinear boundary value problem

$$-\Delta u - \lambda \exp(u) = 0 \quad \text{in} \quad S, \tag{5.1}$$

$$u = 0 \quad \text{on} \quad \partial S \tag{5.2}$$

as a function of the parameter λ, where Δ denotes the Laplacian, S the unit square, and ∂S its boundary. This problem is known as the Bratu problem and is a common test problem for path following methods. We discretize S by a uniform grid with $(\ell-1)^2$ interior grid points (s_k, t_k), where $t_k = s_k = k/\ell$, $1 \leq k < \ell$, and approximate the Laplacian by the standard five-point stencil. This yields a system of $(\ell-1)^2$ nonlinear equations

$$G(w, \lambda) = 0, \tag{5.3}$$

where the entries of the vector $w \in \mathbb{R}^{(\ell-1)^2}$ are approximations of the function u at the grid points. Numerous techniques for computing $w(\lambda)$ as λ is increased from, say, λ_0 to λ_1 are available; see, e.g., [1, 5, 6] and the references therein.

The matrix $\partial G/\partial w$ is singular at turning points (w, λ) of the path $\lambda \to (w(\lambda), \lambda)$, and one often introduces an auxiliary parameter η in order to be able to traverse these points. Thus, let $\lambda = \lambda(\eta)$ and assume that $w(\lambda(\hat{\eta}))$ is available, where

$\lambda_0 \leq \lambda(\hat{\eta}) \leq \lambda_1$. We would like to determine $\lambda(\hat{\eta} + \delta\eta)$ and $w(\lambda(\hat{\eta} + \delta\eta))$. Introduce the function

$$L(w, \lambda, \delta\eta) = d^*(w - w(\lambda(\hat{\eta}))) + c(\lambda - \lambda(\hat{\eta})) - \delta\eta \tag{5.4}$$

for some $d \in \mathbb{R}^{(\ell-1)^2}$ and $c \in \mathbb{R}$. The choice of d and c will be commented on below. Let $(w^{(j)}, \lambda^{(j)})$ be an available approximation of the solution of

$$\begin{aligned} G(w, \lambda) &= 0, \\ L(w, \lambda, \delta\eta) &= 0. \end{aligned} \tag{5.5}$$

Newton's method can be used to determine an improved approximation

$$(w^{(j+1)}, \lambda^{(j+1)}) = (w^{(j)} + \delta w, \lambda^{(j)} + \delta\lambda)$$

of the solution $(w(\lambda(\hat{\eta} + \delta\eta)), \lambda(\hat{\eta} + \delta\eta))$ of (5.5), where δw and $\delta\lambda$ satisfy

$$\begin{bmatrix} G_w^{(j)} & G_\lambda^{(j)} \\ d^* & c \end{bmatrix} \begin{bmatrix} \delta w \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} -G^{(j)} \\ -L^{(j)} \end{bmatrix}, \tag{5.6}$$

with

$$\begin{aligned} G^{(j)} &= G(w^{(j)}, \lambda^{(j)}), & L^{(j)} &= L(w^{(j)}, \lambda^{(j)}, \delta\eta), \\ G_w^{(j)} &= \tfrac{\partial}{\partial w} G(w^{(j)}, \lambda^{(j)}), & G_\lambda^{(j)} &= \tfrac{\partial}{\partial \lambda} G(w^{(j)}, \lambda^{(j)}). \end{aligned}$$

The vector d should be chosen to make the matrix in (5.6) nonsingular even when G_w is singular. This allows simple turning points to be traversed. The parameter η is sometimes chosen to be arc length or pseudo–arc length of the curve $\lambda \to (w(\lambda), \lambda)$. The quantities d, c in (5.4) then may be defined by, e.g., $d = dw(\lambda(\hat{\eta}))/d\eta$, $c = d\lambda(\hat{\eta})/d\eta$.

To illustrate the performance of Algorithm 4.4, we discretize (5.1) on a uniform grid with $\ell = 26$. The matrix in (5.6) then is of size 626×626. We choose $\lambda = \exp(\eta) - 1$ and seek to determine the solution of (5.5) with $\delta\eta = 10$, starting with $w^{(0)} = 0$ and $\lambda^{(0)} = 0$, i.e., $x_0 = 0$ in Algorithm 4.4. Then $G_w^{(0)}$ is the negative discrete Laplacian, $G_\lambda^{(0)} = -[1, 1, \ldots, 1]^*$, $G^{(0)} = 0$, and $L^{(0)} = -\delta\eta$. We let $c = 1$ and, since $\partial w/\partial\eta$ is the largest at the center of the unit square, we choose $d = e_{(\ell-1)^2/2}$. This defines the matrix in (5.6), which we will refer to as A. It has skew-symmetric part of rank $s = 2$; cf. (1.1). We choose

$$f_1 = [1, 1, \ldots, 1, 0]^* - e_{(\ell-1)^2/2}, \quad f_2 = e_{(\ell-1)^2+1}, \quad g_1 = f_2, \quad g_2 = -f_1$$

in the computations.

Algorithm 4.4 reduces the residual error from 10 ($= |\delta\eta|$) to $1.84 \cdot 10^{-7}$ in 50 iterations. In the present example, the numerical values of $\|b - Ax_{50}\|$, $\|r_{50}\|$ as computed by (4.10), and $|\gamma_{50}|$ agree to at least five significant digits. Solution of (5.6) by a direct method gave x_{direct} with $\|x_{\text{direct}} - x_{50}\| = 1.42 \cdot 10^{-10}$. Let x'_{50} denote the approximate solution determined by standard GMRES,[1] and let $r'_{50} = b - Ax'_{50}$. Then $\|r'_{50}\| = 1.84 \cdot 10^{-7}$, $\|x_{\text{direct}} - x'_{50}\| = 1.42 \cdot 10^{-10}$, and $\|x'_{50} - x_{50}\| = 4.90 \cdot 10^{-12}$.

[1]Standard GMRES refers to the commonly used GMRES implementation based on the Arnoldi process with orthogonalization of the Arnoldi vectors by the modified Gram–Schmidt method.

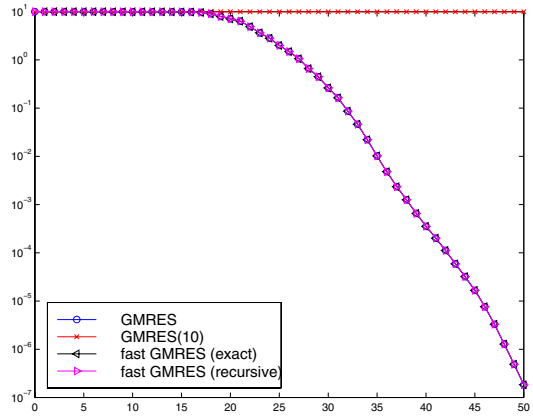


FIG. 5.1. *Residual norms for Algorithm* 4.4 *applied to the data of Example* 5.1. *For comparison, we show both the norm of the exact residuals* $\|b - Ax_k\|$ *(symbol* $\triangleleft$*) and the recursively computed residual norms* $|\gamma_k|$ *(symbol* $\triangleright$*), as well as the norm of the residuals* r'_k *(symbol* $\circ$*) obtained by standard GMRES, which are all of the same size. In contrast, restarted GMRES*(10) *(symbol* $\times$*) fails to converge.*

Figure 5.1 shows the residual errors for standard GMRES and Algorithm 4.4. Let x_k denote the iterates computed by Algorithm 4.4 and let γ_k be the recursively evaluated quantities in the algorithm, such that (in exact arithmetic) $|\gamma_k| = \|b - Ax_k\|$. Figure 5.1 displays $|\gamma_k|$, referred to as [illegible], as well as the evaluated norms $\|b - Ax_k\|$, referred to as [illegible], for $0 \le k \le 50$. The $|\gamma_k|$ are seen to be accurate approximations of $\|b - Ax_k\|$. Moreover, the latter quantities are of the same size as the residual norms produced by standard GMRES.

Convergence is slow during the first 15 iterations and can be sped up by the use of a preconditioner. Note that a symmetric positive definite preconditioner would not change the rank of the skew-symmetric part.

Algorithm 4.4 requires about the same computer storage as GMRES restarted every $2s+6$ iterations. The latter method is referred to as restarted GMRES$(2s+6)$. We also compare Algorithm 4.4 to restarted GMRES$(2s+6)$. For the present example restarted GMRES$(2s+6)$ with $s = 2$ fails to converge; see Figure 5.1.

Both standard and restarted GMRES are implemented using modified Gram–Schmidt orthogonalization of the Arnoldi vectors. Algorithm 4.4 explicitly orthogonalizes each new Arnoldi vector v_{k+1} only against the two most recently generated vectors, v_k and v_{k-1}. Therefore, the orthogonality properties of the matrices $V_k = [v_1, v_2, \ldots, v_k]$ determined by standard GMRES and Algorithm 4.4 in finite precision arithmetic may differ. Figure 5.2 displays the quantities $\|I_k - V_k^* V_k\|^2$, for $1 \le k \le 50$, for matrices V_k determined by standard GMRES and Algorithm 4.4. In this example, the columns of the matrices V_k determined by Algorithm 4.4 are closer to orthonormal than those determined by standard GMRES.

[illegible] 5.2. The integral equation

$$\gamma\, u(\alpha) + \frac{1}{\pi} \int_{-1}^{1} \frac{d}{d^2 + (\alpha - \beta)^2} u(\beta) d\beta = f(\alpha), \qquad -1 \le \alpha \le 1, \tag{5.7}$$

with $\gamma = 1$ and d a positive constant, is known as Love's integral equation. It arises in electrostatics; see, e.g., Baker [3, p. 258]. Let $f(\alpha) = (1 + \alpha)^{1/2}$, let $d = 1/10$, and discretize (5.7) by a Nyström method based on the composite trapezoidal rule with

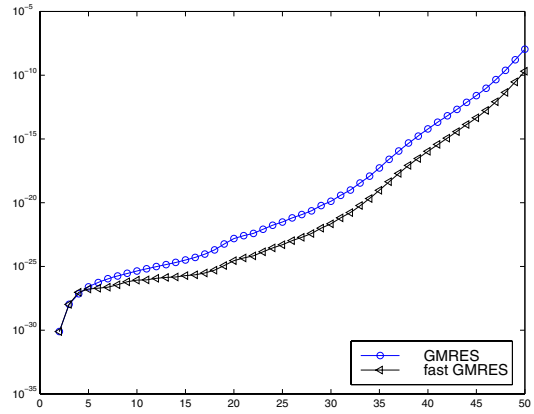


FIG. 5.2. *Orthonormality of the Arnoldi vectors for Example* 5.1: $\|I_k - V_k^* V_k\|^2$ *as a function of* k *for Algorithm* 4.4 *(symbol* $\triangleleft$*) and standard GMRES (symbol* $\circ$*).*

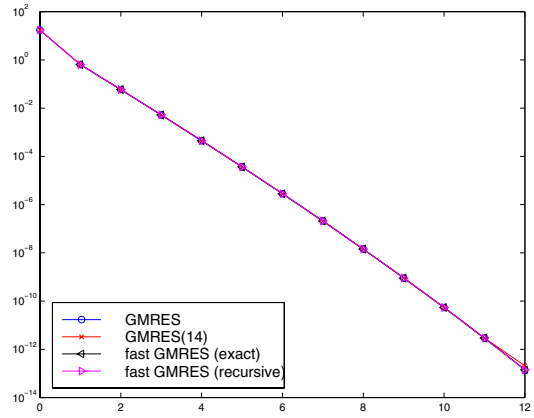


FIG. 5.3. *Residual norms for Algorithm* 4.4 *applied to the data of Example* 5.2. *For comparison, we show both the norm of the exact residuals* $\|b - Ax_k\|$ *(symbol* $\triangleleft$*) and the recursive residual norms* $|\gamma_k|$ *(symbol* $\triangleright$*), which are of the same size. The norm of the residuals* r'_k *obtained by standard GMRES (symbol* $\circ$*) and by restarted GMRES*(14) *(symbol* $\times$*) are also displayed.*

equidistant nodes $\alpha_k = \beta_k = (k-1)/(n-1)$, $1 \le k \le n$, $n = 300$. This gives a linear system of equations with a matrix of the form

$$A = \gamma I + KD, \tag{5.8}$$

where K is a symmetric Toeplitz matrix and $D = \mathrm{diag}[1/2, 1, 1, \ldots, 1, 1/2]$. The skew-symmetric part of A therefore is of rank $s = 4$. The memory requirement of Algorithm 4.4 is about the same as for restarted GMRES(14).

Figure 5.3 shows the residual errors for Algorithm 4.4 as given by $|\gamma_k|$ and $\|b - Ax_k\|$ for $0 \le k \le 12$, as well as the corresponding residual errors for standard GMRES. The initial approximate solution is $x_0 = 0$. The iterations are terminated as soon as the residual error for standard GMRES is of norm smaller than $1 \cdot 10^{-12}$. Convergence is rapid both for Algorithm 4.4 and standard GMRES, and the methods produce iterates with residual errors of nearly the same size.

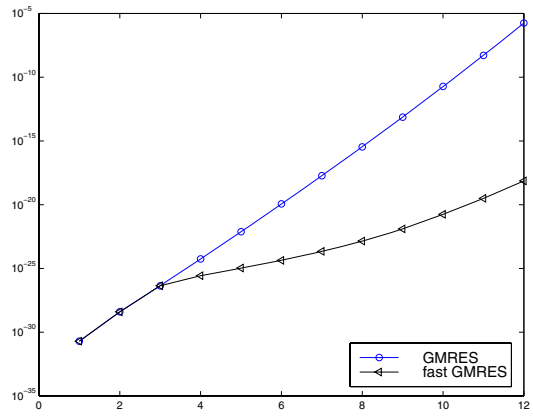


FIG. 5.4. *Orthonormality of the Arnoldi vectors for Example* 5.2: $\|I_k - V_k^* V_k\|^2$ *as a function of* k *for Algorithm* 4.4 *(symbol* $\triangleleft$*) and standard GMRES (symbol* $\circ$*).*

Figure 5.4 is analogous to Figure 5.2 and shows that the Arnoldi vectors generated by Algorithm 4.4 are slightly closer to being orthonormal than the Arnoldi vectors determined by standard GMRES.

The nonsymmetric matrix KD in (5.8) is the discretization of a compact integral operator. It has many eigenvalues close to the origin. Therefore the matrix (5.8) has many eigenvalues close to γ, which has the value one in Example 5.2. In the following examples, we will reduce γ. This reduces the rate of convergence and illustrates that, differently from Examples 5.1 and 5.2, the Arnoldi vectors determined by Algorithm 4.4 may be less close to orthonormal than the Arnoldi vectors determined by the Arnoldi process in the standard GMRES implementation.

EXAMPLE 5.3. We modify the integral equation (5.7) of Example 5.2 by setting $\gamma = 0.1$. This change of γ reduces the rate of convergence. Discretization is carried out in the same manner as in Example 5.2. We use the same initial approximate solution and stopping criterion as in Example 5.2.

Figure 5.5 displays the norm of the residual errors for Algorithm 4.4, standard GMRES, and restarted GMRES(14) and is analogous to Figure 5.3. Figure 5.5 shows the residual errors r_{21} and r_{22} determined by Algorithm 4.4 to be of slightly larger norm than the corresponding residual errors determined by standard GMRES. The cause for this can be found in Figure 5.6(a), which shows the quantities $\|I_k - V_k^* V_k\|^2$ for $1 \leq k \leq 22$. The figure shows the Arnoldi vectors computed by Algorithm 4.4 to be slightly less close to orthonormal than are the Arnoldi vectors determined by standard GMRES.

Figure 5.6(b) displays $\|I_{m+1} - V_{m-k:k}^* V_{m-k:k}\|^2$ as a function of k for $m = 1, 2, \ldots, 5$, thus measuring the orthonormality between the last $m+1$ Arnoldi vectors computed by Algorithm 4.4. Orthonormality is lost fairly rapidly for $m \geq 3$.

EXAMPLE 5.4. We modify the integral equation (5.7) of Examples 5.2 and 5.3 by setting $\gamma = 0.01$. This change of γ reduces the rate of convergence compared with Example 5.3. Discretization is carried out in the same manner as in Examples 5.2 and 5.3, and we use the same initial approximate solution and stopping criterion as in those examples.

Figure 5.7 displays the norm of the residual errors for Algorithm 4.4, standard GMRES, and restarted GMRES(14) and is analogous to Figure 5.5. Figure 5.7 shows

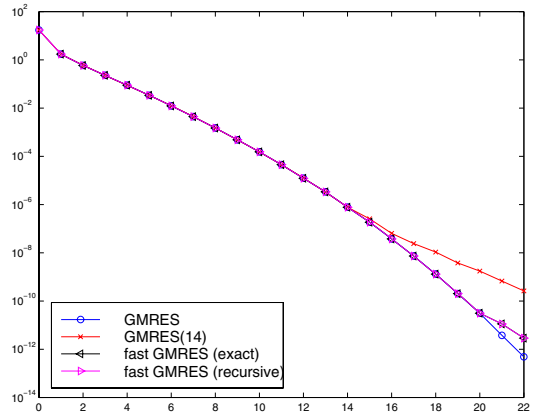


FIG. 5.5. *Residual norms for Algorithm* 4.4 *applied to the data of Example* 5.3. *For comparison, we display both the norm of the exact residuals* $\|b - Ax_k\|$ *(symbol* $\triangleleft$*) and the recursive residual norms* $|\gamma_k|$ *(symbol* $\triangleright$*), which are of the same size, and slightly smaller than those obtained for restarted GMRES*(14) *(symbol* $\times$*). The norms of the residuals* r'_k *determined by standard GMRES (symbol* $\circ$*) are somewhat smaller for* $k \geq 21$.

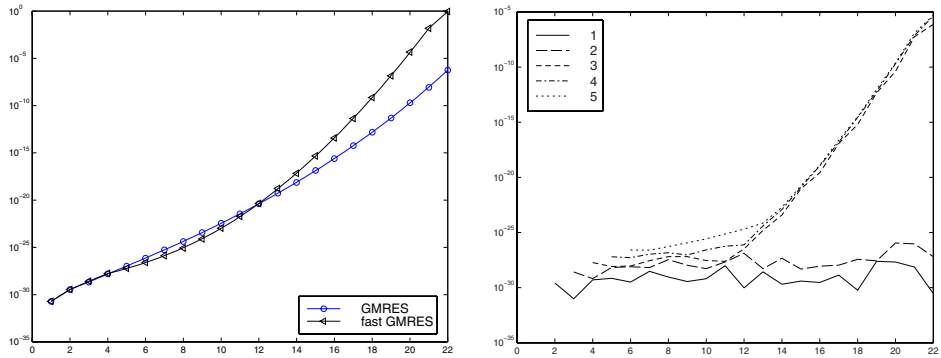


FIG. 5.6. *Orthonormality of the Arnoldi vectors for Example* 5.3: (a) $\|I_k - V_k^* V_k\|^2$ *as a function of* k *for Algorithm* 4.4 *(symbol* $\triangleleft$*) and standard GMRES (symbol* $\circ$*).* (b) *From bottom to top,* $\|I_{m+1} - V_{m-k:k}^* V_{m-k:k}\|^2$ *as a function of* k *for* $m = 1, 2, \ldots, 5$ *for Algorithm* 4.4.

Algorithm 4.4 to reduce the norm of the residual error slower than standard GMRES, but faster than restarted GMRES(14).

The reason for the slower convergence of Algorithm 4.4 is the loss of orthonormality of the Arnoldi vectors generated by the algorithm. The latter is illustrated by Figures 5.8.

Examples 5.3 and 5.4 illustrate that the iterates determined by Algorithm 4.4 may converge slower to the solution than the iterates determined by standard GMRES. A reason for this appears to be that the Arnoldi vectors generated by Algorithm 4.4 may be far from orthonormal; see Example 5.4. The loss of orthogonality and its effect on the convergence of GMRES has received considerable attention in the literature; see, e.g., [8, 10, 13, 14, 16, 17]. For instance, Simoncini and Szyld [16] recently pointed out that loss of orthogonality does not prevent a near-optimal rate of convergence,

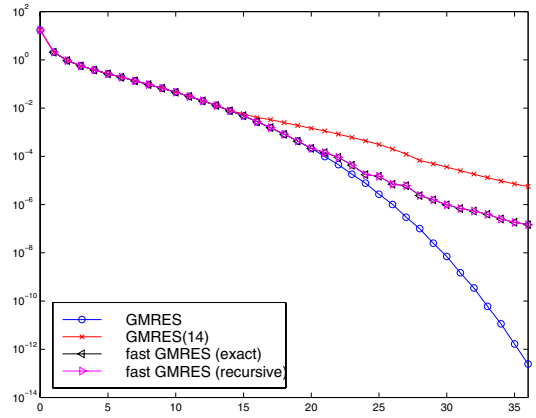


FIG. 5.7. *Residual norms for Algorithm* 4.4 *applied to the data of Example* 5.3. *For comparison, we show both the norm of the exact residuals* $\|b - Ax_k\|$ *(symbol* $\triangleleft$*) and the recursive residual norms* $|\gamma_k|$ *(symbol* $\triangleright$*), which are of the same size, and smaller than those obtained by restarted GMRES*(14) *(symbol* $\times$*). The norms of the residuals* r'_k *obtained by the standard GMRES (symbol* $\circ$*) are much smaller for* $k \geq 30$.

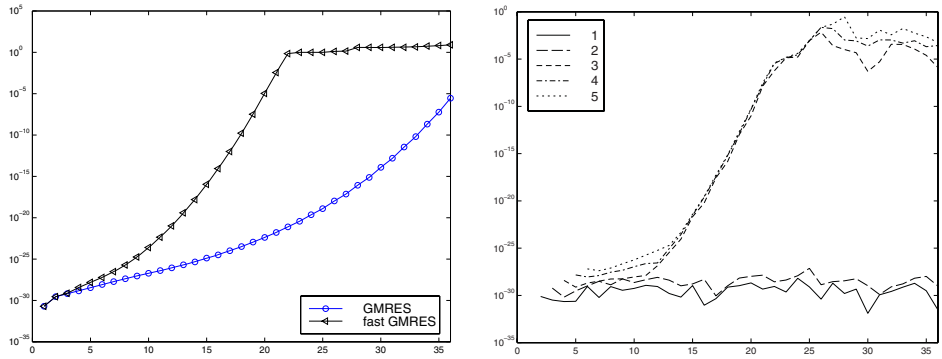


FIG. 5.8. *Orthonormality of the Arnoldi vectors for Example* 5.3: (a) $\|I_k - V_k^* V_k\|^2$ *as a function of* k *for Algorithm* 4.4 *(symbol* $\triangleleft$*) and standard GMRES (symbol* $\circ$*).* (b) *From bottom to top,* $\|I_{m+1} - V_{m-k:k}^* V_{m-k:k}\|^2$ *as a function of* k *for* $m = 1, 2, \ldots, 5$ *for Algorithm* 4.4.

provided that each new Arnoldi vector generated has a sufficiently large angle with the space spanned by the already available Arnoldi vectors. Example 5.4 suggests that the loss of orthogonality also may reduce this angle.

6. Conclusion. Linear systems of equations with a matrix that satisfies (1.1) with a small value of s arise in a variety of applications. For many, but not all, linear systems of equations of this kind, Algorithm 4.4 converges like standard GMRES, but requires less computer storage and arithmetic work. In all our experiments, Algorithm 4.4 converges faster than restarted GMRES($2s+6$), which demands roughly the same amount of computer storage as Algorithm 4.4.

Acknowledgment. We would like to thank a referee for comments.

REFERENCES

[1] E. L. Allgower and K. Georg, *Numerical Continuation Methods*, Springer, Berlin, 1990.

[2] G. Arnold, N. Cundy, J. van den Eshof, A. Frommer, S. Krieg, Th. Lippert, and K. Schäfer, *Numerical methods for the QCD overlap operator* II: *Optimal Krylov subspace methods*, in QCD and Numerical Analysis III, A. Boricci, A. Frommer, B. Joó, A. Kennedy, and B. Pendleton, eds., Lect. Notes Comput. Sci. Eng. 47, Springer, Berlin, 2005, pp. 153–167.

[3] C. T. H. Baker, *Numerical Treatment of Integral Equations*, Clarendon Press, Oxford, 1977.

[4] T. Barth and T. Manteuffel, *Multiple recursion conjugate gradient algorithms, part* I: *Sufficient conditions*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 768–796.

[5] D. Calvetti and L. Reichel, *Iterative methods for large continuation problems*, J. Comput. Appl. Math., 123 (2001), pp. 217–240.

[6] C.-S. Chen, N.-H. Lu, and Z.-L. Weng, *Conjugate gradient methods for continuation problems* II, J. Comput. Appl. Math., 62 (1995), pp. 197–216.

[7] P. Concus and G. H. Golub, *A generalized conjugate gradient method for nonsymmetric systems of linear equations*, in Computing Methods in Applied Science and Engineering, R. Glowinski and J. L. Lions, eds., Springer, New York, 1976, pp. 56–65.

[8] J. Drkosova, A. Greenbaum, M. Rozloznik, and Z. Strakos, *Numerical stability of GMRES*, BIT, 35 (1995), pp. 309–330.

[9] Yu. Eidelman, I. Gohberg, and V. Olshevsky, *The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order*, Linear Algebra Appl., 404 (2005), pp. 305–324.

[10] A. Greenbaum, M. Rozloznik, and Z. Strakos, *Numerical behavior of the modified Gram-Schmidt GMRES implementation*, BIT, 37 (1997), pp. 706–719.

[11] C. Jagels and L. Reichel, *The isometric Arnoldi process and an application to iterative solution of large linear systems*, in Iterative Methods in Linear Algebra, R. Beauwens and P. de Groen, eds., Elsevier, Amsterdam, 1992, pp. 361–369.

[12] C. Jagels and L. Reichel, *A fast minimal residual algorithm for shifted unitary matrices*, Numer. Linear Algebra Appl., 1 (1994), pp. 555–570.

[13] C. C. Paige, M. Rozložník, and Z. Strakoš, *Modified Gram–Schmidt (MGS), least squares, and backward stability of MGS-GMRES*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 264–284.

[14] C. C. Paige and Z. Strakoš, *Residual and backward error bounds in minimum residual Krylov subspace methods*, SIAM J. Sci. Comput., 23 (2002), pp. 1898–1923.

[15] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[16] V. Simoncini and D. B. Szyld, *The effect of non-optimal bases on the convergence of Krylov subspace methods*, Numer. Math., 100 (2005), pp. 711–733.

[17] Z. Strakoš and J. Liesen, *On numerical stability in large scale linear algebraic computations*, Z. Angew. Math. Mech., 85 (2005), pp. 307–325.

[18] O. Widlund, *A Lanczos method for a class of nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 15 (1978), pp. 801–812.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 121–141

GRAPH CLUSTERING VIA A DISCRETE UNCOUPLING PROCESS*

STIJN VAN DONGEN†

Abstract. A discrete uncoupling process for finite spaces is introduced, called the *Markov Cluster Process* or the *MCL process*. The process is the engine for the graph clustering algorithm called the *MCL algorithm*. The *MCL process* takes a stochastic matrix as input, and then alternates expansion and inflation, each step defining a stochastic matrix in terms of the previous one. Expansion corresponds with taking the kth power of a stochastic matrix, where $k \in \mathbb{N}$. Inflation corresponds with a parametrized operator Γ_r, $r \geq 0$, that maps the set of (column) stochastic matrices onto itself. The image $\Gamma_r M$ is obtained by raising each entry in M to the rth power and rescaling each column to have sum 1 again. In practice the process converges very fast towards a limit that is invariant under both matrix multiplication and inflation, with quadratic convergence around the limit points. The heuristic behind the process is its expected behavior for (Markov) graphs possessing cluster structure. The process is typically applied to the matrix of random walks on a given graph G, and the connected components of (the graph associated with) the process limit generically allow a clustering interpretation of G. The limit is in general extremely sparse and iterands are sparse in a weighted sense, implying that the *MCL algorithm* is very fast and highly scalable. Several mathematical properties of the *MCL process* are established. Most notably, the process (and algorithm) iterands posses structural properties generalizing the mapping from process limits onto clusterings. The inflation operator Γ_r maps the class of matrices that are diagonally similar to a symmetric matrix onto itself. The phrase *diagonally positive semi-definite* (*dpsd*) is used for matrices that are diagonally similar to a positive semi-definite matrix. For $r \in \mathbb{N}$ and for M a stochastic *dpsd* matrix, the image $\Gamma_r M$ is again *dpsd*. Determinantal inequalities satisfied by a *dpsd* matrix M imply a natural ordering among the diagonal elements of M, generalizing the mapping of process limits onto clusterings. The spectrum of $\Gamma_\infty M$ is of the form $\{0^{n-k}, 1^k\}$, where k is the number of endclasses of the ordering associated with M, and n is the dimension of M. This attests to the uncoupling effect of the inflation operator.

Key words. stochastic uncoupling, graph clustering, Markov graph, Markov matrix, diagonal similarity, positive semi-definite matrices, circulant matrices

AMS subject classifications. 68R10, 05C85, 05C90

DOI. 10.1137/040608635

1. Introduction.

The subject of study is a parametrized algebraic process called the Markov Cluster Process (, process), which is the engine of a cluster algorithm for graphs, accordingly named the , algorithm. The algorithm is nothing more than a shell in which parameters are set, the , process is computed, and the result is interpreted. The process itself is defined on the space of stochastic matrices. Given a graph G, the algorithm employs the process by applying it to the matrix of random walks on G.

The , algorithm [11, 12] was first applied in the field of protein family detection [18]. In this setting, proteins are nodes in a graph where the edge weights are derived from BLAST (Basic Local Alignment Search Tool) scores between protein amino-acid sequences. Following [18], the algorithm has been widely applied in bioinformatics, in a diversity of settings and applications.

*Received by the editors May 19, 2004; accepted for publication (in revised form) by D. A. Bini July 31, 2007; published electronically February 20, 2008.
http://www.siam.org/journals/simax/30-1/60863.html

†Wellcome Trust Genome Campus, The Wellcome Trust Sanger Institute, Hinxton CB10 1SA Cambridge, United Kingdom (svd@sanger.ac.uk). This author's research was carried out at the CWI, the National Research Institute for Mathematics and Computer Science in the Netherlands.

A number of publications have used , for large scale single species or cross-species protein and gene family analysis, e.g., [15, 16, 28, 38, 52, 60]. Other protein-related , applications in bioinformatics are large scale sequence space analysis [19, 37], hybrid , /single-link clustering [29], orthologous groups [41], kinase proteins [24], secreted proteins [7], eye proteins [39], mobile genetic elements [40], protein interaction networks [5, 54], and protein function determination [61]. Additionally, , has been applied in corpus linguistics [13, 14, 25], content-based image retrieval [32], peer-to-peer network analysis [57], and social network analysis [46].

Factors aiding the adoption of the , algorithm include (a) It generates well-balanced flat (nonhierarchic) clusterings. (b) It is intrinsically a bootstrapping method. Seeding information cannot and need not be supplied, especially not the number of clusters. (c) It has a natural parameter (,) affecting cluster granularity. (d) It is amenable to sparse graph/matrix implementation techniques, implying good scalability. (e) Mathematical results tie , process iterands, the cluster interpretation, inflation, and the number of clusters together.

The focus of the present work is largely on (e), the mathematical results describing in a qualitative manner how the , process exposes cluster structure in graphs. Issues of scaling and implementation are discussed, and in two examples the , process and its clustering characteristics are visualized. Relationships with other mathematical frameworks are established, and several conjectures are made. Comparison with other clustering approaches fall outside the scope of this exposition. The field of bioinformatics is very active in this respect, and the reader is referred to the references given above.

The , process is simple to compute and lends itself to drastic scaling by a regime of pruning, as the limits are in general extremely sparse and the iterands sparse in a weighted sense. It is convenient to distinguish between the process and the algorithm, in order to separate mathematical issues from such issues as implementation and scaling (i.e., computing an approximated process in order to gain speed). Section 6 contains a succinct discussion of how an , implementation can efficiently compute a slightly perturbed , process.

The structure of the article is as follows. The clustering heuristic is briefly introduced in the next section. The , process is fully described and the interpretation of a process limit as a clustering of the input graph is given. This is sufficient to define the , algorithm. A summary is given of some issues concerning convergence and the interpretation of limits as clusterings. Several matrix excerpts from one particular process are shown in section 3, including its limit. In section 4 various lemmas and theorems concerning , iterands are given. The process consists of alternation of two operators, expansion, and inflation. Both operators preserve the class of stochastic matrices that are diagonally similar to a symmetric matrix. These matrices are called . Several of their properties are listed. If a matrix is diagonally similar to a positive semi-definite matrix, then it is called a , abbreviated . Under certain weak conditions many iterands are guaranteed to be . Section 5 introduces structure theory for matrices. Such a matrix possesses structural properties inducing a canonical mapping from the matrix onto a directed acyclic graph, generalizing the mapping from , limits onto overlapping clusterings. The structure theory also yields a qualitative statement on the working of the inflation operator in terms of the matrix spectrum. Implementation is discussed in section 6, and conclusions, further research, and related research make up the last section.

2. Preliminaries. The process consists of alternation of matrix expansion and matrix inflation, where expansion means taking the power of a matrix using the usual matrix product, and inflation (denoted Γ_r) means taking the Hadamard power with coefficient r of a stochastic matrix and subsequently scaling its columns to have sum 1 again. The clustering heuristic associated with the process is that a dense region in a graph corresponds with a node set S for which pairs of elements in S have the property that there are relatively many higher length paths completely contained in S itself. By matrix expansion the higher step transition probabilities are obtained; by matrix inflation large probabilities are promoted, and small probabilities are demoted. It is to be expected that probabilities that correspond with edges connecting different dense regions will suffer the most from the process of alternating expansion and inflation. Indeed, iteration of the two operators leads to a limit that is meaningful considering the original heuristic.

The inflation operator Γ_r is defined for arbitrary nonnegative matrices, in a columnwise manner. This implies that column stochastic matrices will be used rather than row stochastic matrices, which is merely a matter of preference and convention. There are no restrictions on the matrix dimensions to fit a square matrix, because this allows Γ_r to act on both matrices and vectors. There is no restriction that the input matrices be stochastic, since it is not strictly necessary, and the extended applicability is sometimes useful. Following the terminology used in [8] and [27], a nonnegative matrix is called if all its columns have at least one nonzero entry. The next definition prepares for the definition of the process.

DEFINITION 2.1. tth Exp_t $\mathrm{Exp}_t A = A^t$

This definition is put in such general terms because the class of dpsd matrices (to be introduced later) allows the introduction of fractional matrix powers in a well-defined way.

DEFINITION 2.2. r $M \in \mathbb{R}_{\geq 0}{}^{m \times n}$ M Γ_r

$$(\Gamma_r M)_{pq} \;=\; (M_{pq})^r \,/\, \sum_{i=1}^{m} (M_{iq})^r.$$

In the setting of the process, positive values r have a sensible interpretation attached to them. Values of r between 0 and 1 increase the homogeneity of the argument probability vector (matrix), whereas values of r between 1 and ∞ increase the inhomogeneity. In both cases, the ordering of the probabilities is not disturbed. Negative values of r invert the ordering, which is not of apparent use. With $\otimes$ denoting the Kronecker product, the identities $\mathrm{Exp}_r(A \otimes B) = \mathrm{Exp}_r(A) \otimes \mathrm{Exp}_r(B)$ and $\mathrm{Exp}_r(\mathrm{Exp}_s(A)) = \mathrm{Exp}_{rs}(A)$ hold. Similarly, $\Gamma_r(A \otimes B) = \Gamma_r(A) \otimes \Gamma_r(B)$ and $\Gamma_r(\Gamma_s(A)) = \Gamma_{rs}(A)$ are true.

DEFINITION 2.3. Γ_∞ $\Gamma_\infty M = \lim_{r \to \infty} \Gamma_r M$

This definition is meaningful, and it is easy to derive the structure of $\Gamma_\infty M$. Each column q of $\Gamma_\infty M$ has k nonzero entries equal to $1/k$, (k depending on q), where k is the number of elements that equal $\max_p M_{pq}$, and the positions of the nonzero entries in $\Gamma_\infty M[1, \ldots, n|q]$ correspond with the positions of the maximal entries in $M[1, \ldots, n|q]$. Following [44], if x denotes a real vector of length n, then $x_{[1]} \geq x_{[2]} \geq \cdots \geq x_{[n]}$ denote the entries of x in decreasing order.

DEFINITION 2.4. [illegible] x, y [illegible] n [illegible] y [illegible] x [illegible] $x \prec y$,

$$\sum_{i=1}^{k} y_{[i]} \geq \sum_{i=1}^{k} x_{[i]} \quad k = 1, \ldots, n, \tag{2.1}$$

$$\sum_{i=1}^{n} y_{[i]} = \sum_{i=1}^{n} x_{[i]}. \tag{2.2}$$

LEMMA 2.5. [illegible] x [illegible] $r, s \in \mathbb{R}_{>0}$, $r < s$ [illegible] $\Gamma_r(x) \prec \Gamma_s(x)$

The proof of this lemma is straightforward [11].

DEFINITION 2.6. [illegible] process [illegible] M [illegible] M [illegible] M [illegible] $e_{(i)}$, $r_{(i)}$ [illegible] $e_i \in \mathbb{N}, e_i > 1$ [illegible] $r_i \in \mathbb{R}, r_i \geq 0$ [illegible]

$$(M, e_{(i)}, r_{(i)}). \tag{2.3}$$

[illegible] $(M, e_{(i)}, r_{(i)})$ [illegible] $M_{(i)}$ [illegible] $M_1 = M$, $M_{2i} = \mathrm{Exp}_{e_i}(M_{2i-1})$ [illegible] $M_{2i+1} = \Gamma_{r_i}(M_{2i})$, $i = 1, \ldots, \infty$

It must be stressed that the [illegible] process has no stochastic interpretation. The heuristic on which it is grounded uses stochastic terminology, but each [illegible] process $(M, e_{(i)}, r_{(i)})$ is (for varying M) really a rather complex dynamical system based on the alternation of two operators, expansion and inflation. The fact that expansion and inflation distribute over the Kronecker yields the following lemma.

LEMMA 2.7. [illegible]

Note. In practice, clustering with the [illegible] algorithm is best done with all expansion values e_i set to two. The reasoning behind this is pragmatic, as inflation can be used to control the mixing properties of the process, whereas expansion is computationally costly. Applying (columnwise) pruning in order to scale the process renders prolonged expansion virtually useless. Nevertheless it seems best to formulate the [illegible] process in the general terms of Definition 2.6, as this supplies a natural framework for questions and conjectures (section 7). The canonical mapping between graphs with nonnegative weights and nonnegative matrices is given below. In order to work with column stochastic matrices, an arbitrary choice is made to identify matrix columns with lists of neighbors.

DEFINITION 2.8. [illegible] associated graph [illegible] A [illegible] n [illegible] n [illegible] $\{1, \ldots, n\}$ [illegible] q, p [illegible] A_{pq}. $A_{pq} > 0$

The following theorem is preparatory to the mapping from nonnegative idempotent matrices to overlapping clusterings in Definition 2.11. Its proof is given in [11] and can also be derived from the decomposition of nonnegative idempotent matrices given in [2, p. 65]. It represents a very basic result on the structural properties of nonnegative idempotent matrices. Theorem 5.4 will show a more general structure to be present in [illegible] iterands, so that in the setting of the [illegible] process Theorem 2.9 becomes a limiting case of Theorem 5.4. It will be shown that for M stochastic [illegible] a finite power of the matrix $\Gamma_\infty(M)$ is idempotent (section 5).

THEOREM 2.9 (see Theorem 1 in [11, p. 18]). [illegible] M [illegible] n [illegible] G [illegible] s, t [illegible] G [illegible] $s \to t$ [illegible] G [illegible] s, t [illegible]

$s \to t \iff M_{ts} \neq 0$ [illegible] α, β, γ [illegible] G [illegible]

$$(\alpha \to \beta) \wedge (\beta \to \gamma) \implies \alpha \to \gamma, \tag{2.4}$$

$$(\alpha \to \alpha) \wedge (\alpha \to \beta) \implies \beta \to \alpha, \tag{2.5}$$

$$\alpha \to \beta \implies \beta \to \beta. \tag{2.6}$$

The theorem basically states that the graph associated with the matrix consists for one part of subgraphs that are complete, with all nodes having loops as well. The other part consists of nodes without loops that, given a complete subgraph, are connected either to all or to none of the nodes in that subgraph. It is convenient to introduce the notions of [illegible] and [illegible]. The second is a (maximal) complete subgraph, and the first is a node in such a subgraph.

Definition 2.10. [illegible] G [illegible] M [illegible] n [illegible] $1, \ldots, n$ [illegible] α [illegible] attractor [illegible] $M_{\alpha\alpha} \neq 0$ [illegible] α [illegible] α [illegible] attractor system

By Theorem 2.9, each attractor system in G induces a weighted subgraph in G that is complete. These subgraphs form the cores of the clustering associated with a (nonnegative idempotent) matrix M as stated below. An attractor system is simply extended with all the nodes that reach it.

Definition 2.11. [illegible] M [illegible] n [illegible] G [illegible] $V = \{1, \ldots, n\}$ [illegible] $E_i, i = 1, \ldots, k$ [illegible] G [illegible] $v \in V$ [illegible] $v \to E_i$ [illegible] $e \in E_i$ [illegible] $v \to e$ [illegible] $\mathcal{C} = \{C_1, \ldots, C_k\}$ [illegible] M [illegible]

$$C_i = E_i \cup \{v \in V \mid v \to E_i\}. \tag{2.7}$$

Theorem 2.9 implies that $v \to f$ for all $f \in E_i$.

The simplest example of a limit matrix inducing overlap is the matrix below, giving rise to the clustering $\{1, 3\}, \{2, 3\}$:

$$\begin{pmatrix} 1 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ 0 & 0 & 0 \end{pmatrix}.$$

Combining the previous simple results, it is possible to rewrite each nonnegative column allowable idempotent matrix M as a form $P^T AP$, where P is a permutation matrix, and

$$A = \begin{pmatrix} B_1 & & & f_{11} & f_{12} & \ldots & f_{1l} \\ & \ddots & & \vdots & \vdots & \vdots & \vdots \\ & & B_k & f_{k1} & f_{k2} & \ldots & f_{kl} \\ 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \end{pmatrix}.$$

Each matrix B_i is square, has rank one with all columns identical, contains only positive entries, and there are no other nonzero entries in the corresponding columns of A. Each matrix B_i corresponds with an attractor system, and k is the number of resulting clusters. Each f_{ij} is a column vector with the same number of (row) entries as B_i. Either all entries in f_{ij} are zero or they are all nonzero, and for each j at least one f_{ij} is nonzero. If the vector f_{ij} is nonzero, then it corresponds with a node

(identified by j) that is in the cluster defined by the attractor system corresponding with B_i. If f_{ij} is nonzero for more than one i, then those i determine clusters that overlap in the node identified by j.

In practice, cluster overlap is very rare. The phenomenon is inherently unstable, in the sense that applying the process to a perturbation of a limit matrix that induces overlap leads the process to converge to a limit no longer inducing overlap. A node previously in overlap will then be associated with just one of the multiple clusters it was associated with before [12]. All current evidence suggests that cluster overlap implies the existence of a graph automorphism of the graph associated with the input matrix, leaving the overlapping part invariant and mapping the overlapping clusters onto each other. In the simple example above, the automorphism would send $(1, 2, 3)$ to $(2, 1, 3)$.

The phenomenon of attractor systems of cardinality greater than one is also unstable in nature, but a small perturbation of a matrix limit having such a system will not change the associated clustering (assuming that the parameter r of Γ_r is bounded). The main reason for this is that if J is a stochastic matrix of rank one and E is a perturbation matrix (with zero column sums) of sufficiently small norm, then (restricting attention to a special case) $\mathrm{Exp}_2(\Gamma_2(J + E))$ is of the form $J' + E'$, with J' stochastic of rank one and the norm of E' being of order square the norm of E. Current evidence also suggests that attractor systems of cardinality greater than one imply the existence of a set of automorphisms by which each of the attractors (of one system) can be mapped to any of the other. An example is shown in Figure 3.2 for the graph in Figure 3.1. In this case, the automorphism would leave all nodes in place except for interchanging 9 and 11.

Assuming that e_i equals two and r_i is bounded eventually, it is true that the process converges quadratically in the neighborhood of matrices that (i) are , that is, invariant both under expansion (multiplication) and inflation, and (ii) have in each column one entry equal to 1 and all other entries equal to 0. This is straightforward (though tedious) to verify—proofs are given in [11]. The issue is somewhat clouded by the fact that the process may also converge towards a limit matrix that does not satisfy condition (ii). A small perturbation of such a matrix is amplified by the inflation operator so that the sequence of iterands departs from it.

The algorithm consists of three steps. First, given an arbitrary input graph G, loops are added resulting in a graph informally denoted as $G + \Delta$. Some remarks on the necessity of this step are made in the next section. How weights are chosen for the loops to be added is the responsibility of the algorithm. Subsequently, an process is applied to the matrix of random walks associated with $G + \Delta$. Third, the limit thus computed is interpreted as a clustering according to Definition 2.11. One can obtain a fast, robust, and well-scaling implementation of the algorithm at http://micans.org/mcl/, which allows a simple type of parametrization: The expansion values e_i are all set to 2 and the inflation values r_i can assume two values, changing once from the first to the second value.

In general the limit of an process is extremely sparse, as the inflation operator is a force driving towards sparse columns. iterands tend to be sparse in a weighted sense, and this supplies the means to scale the algorithm drastically by incorporating a regime of pruning into the process (cf. section 6).

The natural way to use the process for the purpose of clustering a graph is

by applying it to the matrix which represents the standard concept of a random walk on the graph, where loops have been added to the graph. This matrix is obtained as the incidence matrix multiplied by the diagonal matrix of inverse column (row) sums, so that the product is column (row) stochastic. If the graph is undirected, then the resulting stochastic matrix is diagonally similar to a symmetric matrix.

3. Examples.

Example I. In Figure 3.2, four excerpts are given of an , process. These are the input matrix M, the iterand $M_3 = \Gamma_2 M^2$, the iterand $M_5 = \Gamma_2(\Gamma_2 M^2 \cdot \Gamma_2 M^2)$, and the stable limit denoted L_M. The process consists entirely of alternation of Exp_2 and Γ_2. The graph H associated with M is depicted in Figure 3.1. Every node in H has a loop; these are all left out in the figure. Weights are omitted as well. Note that there exists a diagonal matrix d such that Md is symmetric. This implies that $d^{-1/2}Md^{1/2}$ is symmetric and thus the spectrum of M is real. Interpreting L_M according to Definition 2.11 yields the clustering $\{\{1,6,7,10\},\{2,3,5\},\{4,8,9,11,12\}\}$. It is necessary to add loops to the nodes before applying , in order to prevent a result reflecting the bipartite characteristics of H. Without adding loops, the resulting , process limit yields the clustering $\{\{1,5,10\},\{2,6,7\},\{3,4,8,9,11,12\}\}$. This is in line with the heuristic underlying the process: The probabilities that are initially boosted correspond with 2-step paths in H.

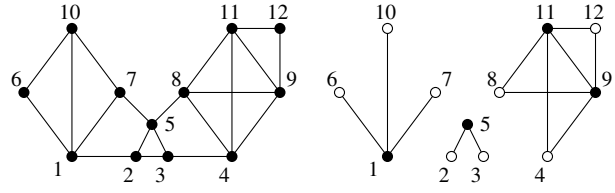


FIG. 3.1. *On the left a graph H, on the right the graph associated with the limit of an MCL process applied to H, loops added to H. Dark circles signify attractors; nodes* 9 *and* 11 *form an attractor system (refer to section* 5*). Compare with the matrix iterands and limit matrix in Figure* 3.2 *and with Figure* 3.3*, and see the discussion in Example* I.

Example II. Figure 3.3 depicts different iterands of an , process triggered by a geometric graph. This graph was first used in [21] as a test case for graph partitioning. It is shown in the upper left of the figure. Two nodes are connected if their distance is at most $\sqrt{8}$ Euclidean units. The edge weights were taken inversely proportional to the Manhattan distance, and loops were added to each node with a weight equal to the largest weight found in the edges in which it participates. The matrix of random walks on this graph was input to an , process in which the sequence $e_{(i)}$ assumed the constant 2 everywhere, and the sequence $r_{(i)}$ assumed the constant 1.3 everywhere.

The other graphs in Figure 3.3 represent a pictorial representation of four , iterands (stochastic matrices) and the limit in the lower right. The degree of shading of a bond between two nodes indicates the maximum value of the corresponding transition probabilities taken over the two directions. The darker the bond, the larger the maximum. The degree of shading of a node indicates the total sum of incoming transition probabilities. Thus, a dark bond between a white node and a black node indicates that the maximum transition probability is found in the direction of the black node, and that the probability attached to the reverse arc is negligible. The limit graph, depicted in the lower right, contains all necessary information needed for

$$\begin{pmatrix}
0.200 & 0.250 & -- & -- & -- & 0.333 & 0.250 & -- & -- & 0.250 & -- & -- \\
0.200 & 0.250 & 0.250 & -- & 0.200 & -- & -- & -- & -- & -- & -- & -- \\
-- & 0.250 & 0.250 & 0.200 & 0.200 & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & 0.250 & 0.200 & -- & -- & -- & 0.200 & 0.200 & -- & 0.200 & -- \\
-- & 0.250 & 0.250 & -- & 0.200 & -- & 0.250 & 0.200 & -- & -- & -- & -- \\
0.200 & -- & -- & -- & -- & 0.333 & -- & -- & -- & 0.250 & -- & -- \\
0.200 & -- & -- & -- & 0.200 & -- & 0.250 & -- & -- & 0.250 & -- & -- \\
-- & -- & -- & 0.200 & 0.200 & -- & -- & 0.200 & 0.200 & -- & 0.200 & -- \\
-- & -- & -- & 0.200 & -- & -- & -- & 0.200 & 0.200 & -- & 0.200 & 0.333 \\
0.200 & -- & -- & -- & -- & 0.333 & 0.250 & -- & -- & 0.250 & -- & -- \\
-- & -- & -- & 0.200 & -- & -- & -- & 0.200 & 0.200 & -- & 0.200 & 0.333 \\
-- & -- & -- & -- & -- & -- & -- & -- & 0.200 & -- & 0.200 & 0.333
\end{pmatrix}$$

M

$$\begin{pmatrix}
0.380 & 0.087 & 0.027 & -- & 0.077 & 0.295 & 0.201 & -- & -- & 0.320 & -- & -- \\
0.047 & 0.347 & 0.210 & 0.017 & 0.150 & 0.019 & 0.066 & 0.011 & -- & 0.012 & -- & -- \\
0.014 & 0.210 & 0.347 & 0.055 & 0.150 & -- & 0.016 & 0.046 & 0.009 & -- & 0.009 & -- \\
-- & 0.027 & 0.087 & 0.302 & 0.062 & -- & -- & 0.184 & 0.143 & -- & 0.143 & 0.083 \\
0.058 & 0.210 & 0.210 & 0.055 & 0.406 & -- & 0.083 & 0.046 & 0.009 & 0.019 & 0.009 & -- \\
0.142 & 0.017 & -- & -- & -- & 0.295 & 0.083 & -- & -- & 0.184 & -- & -- \\
0.113 & 0.069 & 0.017 & -- & 0.062 & 0.097 & 0.333 & 0.011 & -- & 0.147 & -- & -- \\
-- & 0.017 & 0.069 & 0.175 & 0.049 & -- & 0.016 & 0.287 & 0.143 & -- & 0.143 & 0.083 \\
-- & -- & 0.017 & 0.175 & 0.012 & -- & -- & 0.184 & 0.288 & -- & 0.288 & 0.278 \\
0.246 & 0.017 & -- & -- & 0.019 & 0.295 & 0.201 & -- & -- & 0.320 & -- & -- \\
-- & -- & 0.017 & 0.175 & 0.012 & -- & -- & 0.184 & 0.288 & -- & 0.288 & 0.278 \\
-- & -- & -- & 0.044 & -- & -- & -- & 0.046 & 0.120 & -- & 0.120 & 0.278
\end{pmatrix}$$

$\Gamma_2 M^2$

$$\begin{pmatrix}
0.448 & 0.080 & 0.023 & 0.000 & 0.068 & 0.426 & 0.359 & 0.000 & 0.000 & 0.432 & 0.000 & -- \\
0.018 & 0.285 & 0.228 & 0.007 & 0.176 & 0.006 & 0.033 & 0.005 & 0.000 & 0.007 & 0.000 & 0.000 \\
0.005 & 0.223 & 0.290 & 0.022 & 0.173 & 0.000 & 0.010 & 0.017 & 0.003 & 0.001 & 0.003 & 0.001 \\
0.000 & 0.018 & 0.059 & 0.222 & 0.040 & 0.000 & 0.001 & 0.187 & 0.139 & 0.000 & 0.139 & 0.099 \\
0.027 & 0.312 & 0.314 & 0.028 & 0.439 & 0.005 & 0.054 & 0.022 & 0.003 & 0.010 & 0.003 & 0.001 \\
0.116 & 0.007 & 0.001 & 0.000 & 0.004 & 0.157 & 0.085 & 0.000 & -- & 0.131 & -- & -- \\
0.096 & 0.040 & 0.013 & 0.000 & 0.037 & 0.083 & 0.197 & 0.001 & 0.000 & 0.104 & 0.000 & 0.000 \\
0.000 & 0.012 & 0.042 & 0.172 & 0.029 & 0.000 & 0.002 & 0.198 & 0.133 & 0.000 & 0.133 & 0.096 \\
0.000 & 0.001 & 0.015 & 0.256 & 0.009 & -- & 0.000 & 0.266 & 0.326 & 0.000 & 0.326 & 0.346 \\
0.290 & 0.021 & 0.002 & 0.000 & 0.017 & 0.323 & 0.260 & 0.000 & 0.000 & 0.316 & 0.000 & -- \\
0.000 & 0.001 & 0.015 & 0.256 & 0.009 & -- & 0.000 & 0.266 & 0.326 & 0.000 & 0.326 & 0.346 \\
-- & 0.000 & 0.001 & 0.037 & 0.000 & -- & 0.000 & 0.039 & 0.069 & -- & 0.069 & 0.112
\end{pmatrix}$$

$\Gamma_2(\Gamma_2 M^2 \cdot \Gamma_2 M^2)$

$$\begin{pmatrix}
1.000 & -- & -- & -- & -- & 1.000 & 1.000 & -- & -- & 1.000 & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & 1.000 & 1.000 & -- & 1.000 & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & --
\end{pmatrix}$$

Limit L_M resulting from iterating $(\Gamma_2 \circ \mathrm{Exp}_2)$ with initial matrix M, which is the matrix of random walks associated with the graph in Figure 3.1.

Entries marked "−−" are either zero because that is the exact value they assume (this is true for the first two matrices) or because the computed value fell below the machine precision.

FIG. 3.2. *Iteration of* $(\Gamma_2 \circ \mathrm{Exp}_2)$.

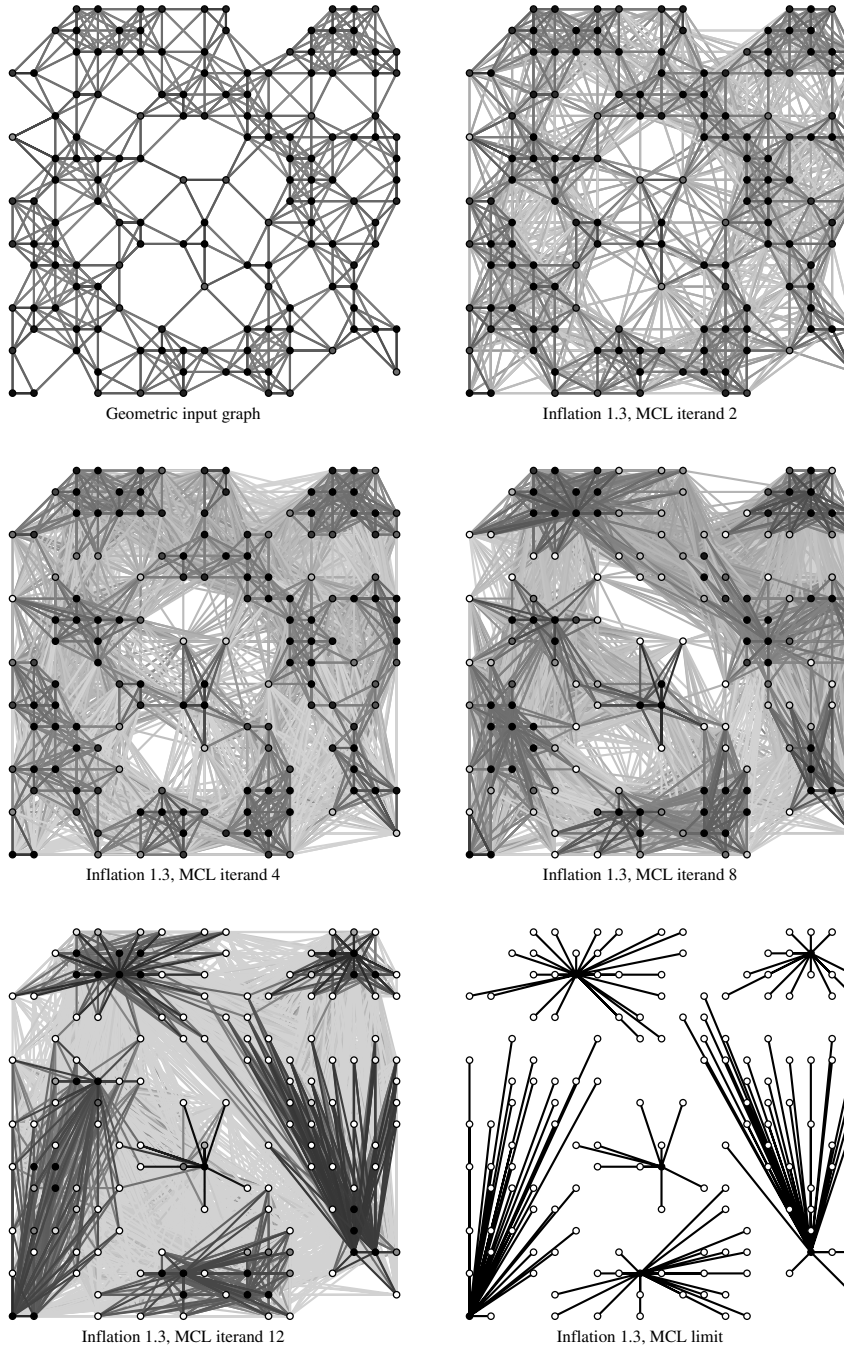


FIG. 3.3. *Visualization of successive stages of the MCL process applied to the upper left graph, with* $e_i = 2$ *and* $r_i = 1.3$ *for every iteration* i *(cf. Definition* 2.6*). The meaning of the grey values of bonds and nodes are explained in section* 3*. At most* 24 *neighbors are shown for each node.*

constructing the , -invariant limit matrix. Dark nodes in this graph are attractors.

The examples in Figures 3.3 and 3.2 indicate that the , process has remarkable convergence properties, regarding the structural properties of its iterands. Considering this evidence, to some extent an analogy is suggested with the normal Markov process.

Assuming that the associated graph of the input matrix M is strongly connected and contains at least one loop, it follows by Perron–Frobenius theory that 1 is the only eigenvalue of M of modulus 1 and that it is simple.

By considering the spectrum of the powers M^k it follows that the normal Markov process converges towards a rank-one idempotent matrix, having spectrum $\{0^{n-1}, 1\}$. In the example shown in Figure 3.2 the process also converges towards an idempotent limit. The multiplicity of its eigenvalue 1 is 3, however, equaling (of course) the number of strongly connected components in the associated graph of the limit. Section 4 will give some insight into the spectral phenomena that play a role in the , process by focusing attention onto two classes of stochastic matrices.

4. Properties of the inflation operator and stochastic *dpsd* matrices. At first sight the inflation operator seems hard to get a grasp on mathematically, though its behavior for vectors is well understood. Lemma 2.5 states that for a stochastic vector x and parameters $r, s \in \mathbb{R}_{>0}$, $r < s$, one has that $\Gamma_r(x) \prec \Gamma_s(x)$, where $\prec$ denotes the majorization relationship. This implies that the orbit $\Gamma_r x$, $(r > 0)$ is fairly well understood, since the limiting cases $\Gamma_r x$, $r \to \infty$ and $\Gamma_r x$, $r \downarrow 0$ are also easily derived. However, majorization results for vectors do not carry over to matrices in such a way that statements can be made about algebraic properties of two matrices subject to a columnwise majorization relationship. In [44] this issue is discussed at length.

To some extent it is possible to give a qualitative account of the behavior of the inflation operator, using structural properties of the matrices in a particular class preserved by inflation. Several preparatory results are derived in the current section. In the following section simple structure theory is developed, explaining the uncoupling effect of the inflation operator in qualitative terms.

In general $\Gamma_r M$ can be described in terms of a Hadamard matrix power that is postmultiplied with a diagonal matrix. For a restricted class of matrices there is an even stronger connection with the Hadamard product. These are the class of stochastic diagonally symmetric matrices and a subclass of the latter, the class of stochastic diagonally positive semi-definite matrices.

The Hadamard (entrywise) product of two matrices A and B that have the same dimensions is written $A \circ B$ and satisfies $[A \circ B]_{pq} = A_{pq} B_{pq}$. The entrywise Hadamard power with exponent r of a matrix A is written $A^{\circ r}$ and satisfies $[A^{\circ r}]_{pq} = {A_{pq}}^r$.

The concept of diagonal symmetrizability can easily be transferred to complex matrices, and most of the results in this paper can be derived in that more general setting. This is not needed in the , setting and hence the definitions and results here are simply stated for real matrices.

DEFINITION 4.1. A diagonally symmetric x Diag$(x)^{-1} A$ Diag(x)

The following useful identity is easy to verify.

LEMMA 4.2. A 4.1 Diag$(x)^{-1} A$ Diag(x) $= [A \circ A^T]^{\circ 1/2}$

DEFINITION 4.3. [illegible] diagonally positive semi-definite [illegible] diagonally positive definite [illegible] dpsd [illegible] dpd [illegible] If M is diagonally symmetric stochastic, and y is such that $M\,\mathrm{Diag}(y)$ is symmetric, then $My = y$; thus y represents the equilibrium distribution of M. In the theory of Markov chains, a stochastic diagonally symmetric matrix is called [illegible] or said to satisfy the [illegible] condition (see, e.g., [43] and [59]). A slightly more general definition and different terminology was chosen here. The main reason is that the term "time reversible" is coupled tightly with the idea of studying a stochastic chain via (powers of) its associated stochastic matrix, and is also used for continuous-time Markov chains. The [illegible] process studied in this article does not have a straightforward stochastic interpretation, and the relationship between an input matrix and the subsequent iterands is much more complex. Moreover, it is natural to introduce the concepts of a matrix being diagonally similar to a positive (semi-) definite matrix; clinging to "time reversible" in this abstract setting would be both contrived and unhelpful. The proposed phrases seem appropriate, since several properties of symmetric and [illegible] matrices remain valid in the more general setting of diagonally symmetric and [illegible] matrices. Lemma 4.4 lists the most important ones, which are easy to verify. Probably all of these results are known.

In the following, submatrices of a matrix A are written $A[u|v]$, where u denotes a list of row indices, and v denotes a list of column indices.

LEMMA 4.4. [illegible] A [illegible] n [illegible] α [illegible] $1,\dots,n$ [illegible] k [illegible] l [illegible] $1,\dots,n$ [illegible] x [illegible] $S = \mathrm{Diag}(x)^{-1} A\,\mathrm{Diag}(x)$ [illegible] $A = \mathrm{Diag}(x) S\,\mathrm{Diag}(x)^{-1}$ [illegible] λ_i [illegible] A ([illegible] S) [illegible] a_i [illegible] A

(a) $A[\alpha|\alpha] = \mathrm{Diag}(x)[\alpha|\alpha]\ \ S[\alpha|\alpha]\ \ \mathrm{Diag}(x)[\alpha|\alpha]^{-1}$ [illegible] A [illegible] S [illegible] A [illegible] A

$$\sum_{i=1}^{k} \lambda_{[i]} \ \geq\ \sum_{i=1}^{k} a_{[i]} \quad k = 1,\dots,n.$$

(b) [illegible] A [illegible] dpsd [illegible] $A_{kk} = 0$ [illegible] k [illegible] k [illegible] A [illegible] A [illegible] dpsd [illegible] $\det A[kl|kl] = 0$ [illegible] k [illegible] l [illegible] k [illegible] l

(c) [illegible] A [illegible] dpsd [illegible] $k \in \mathbb{N}$ [illegible] dpsd [illegible] B [illegible] $B^k = A$ [illegible] $B = \mathrm{Diag}(x) Q \Lambda^{1/k} Q^H \mathrm{Diag}(x)^{-1}$ [illegible] $Q\Lambda Q^H$ [illegible] S [illegible] Λ [illegible] S [illegible] $\Lambda^{1/k}$ [illegible] Λ [illegible] k [illegible] dpsd A [illegible] A^t $t \in \mathbb{R}_{\geq 0}$ [illegible]

(d) A, B n dpsd dpd $A \circ B$ dpsd dpd

. Most statements are easy to verify. For extensive discussion of the majorization relationship between diagonal entries and eigenvalues of symmetric (or hermitian) matrices, as well as results on interlacing inequalities, see [3, 34, 35]. The first statement in (b) follows from the fact that principal minors (of dimension 2) are nonnegative. The second statement can easily be proven by first considering the case where A is symmetric. The determinant $\det A[klm|klm]$ of an extended submatrix equals zero and rewriting the constituent terms yields proportionality as stated in (b). The result for matrices follows trivially. For (c) it is sufficient to use the fact that $Q\Lambda^{1/k}Q^H$ is the unique positive semi-definite kth root of S [34, p. 405]. Statement (d) follows from the identity $(\mathrm{Diag}(x)^{-1}A\ \mathrm{Diag}(x)) \circ (\mathrm{Diag}(y)^{-1}B\ \mathrm{Diag}(y)) = \mathrm{Diag}(x \circ y)^{-1}(A \circ B)\mathrm{Diag}(x \circ y)$ and the fact that the analogous statements for symmetric matrices are true—known under the denomination of [34, p. 458].

. The two most notable properties that do not generalize from symmetric matrices to diagonally symmetric matrices are the absence of an orthogonal basis of eigenvectors for the latter, and the fact that the sum of two diagonally symmetric matrices is in general not diagonally symmetric as well.

Statements (a) and (b) in Lemma 4.4 are used in associating a directed acyclic graph with each matrix in Theorem 5.4. First, the behavior of the inflation operator on diagonally symmetric and matrices is described.

THEOREM 4.5. M n, $\mathrm{Diag}(x)$ $S = \mathrm{Diag}(x)^{-1}M\ \mathrm{Diag}(x)$ r z, $z_k = x_k{}^r(\sum_i M_{ik}{}^r)^{1/2}$ T, $T_{kl} = 1/(\sum_i M_{ik}{}^r)^{1/2}(\sum_i M_{il}{}^r)^{1/2}$

$$\mathrm{Diag}(z)^{-1}\ (\ \Gamma_r M)\ \ \mathrm{Diag}(z) \ \ = \ \ S^{\circ r} \circ T.$$

$\Gamma_r M$

. Define the vector t by $t_k = \sum_i M_{ik}{}^r$. Then

$$\begin{aligned}
\Gamma_r M &= M^{\circ r}\ \mathrm{Diag}(t)^{-1} \\
&= (\mathrm{Diag}(x)\ S\ \mathrm{Diag}(x)^{-1})^{\circ r}\ \mathrm{Diag}(t)^{-1} \\
&= \mathrm{Diag}(x)^{\circ r}\ \ S^{\circ r}\ \ (\mathrm{Diag}(x)^{\circ r})^{-1}\ \mathrm{Diag}(t)^{-1} \\
&= \mathrm{Diag}(t)^{1/2}\mathrm{Diag}(t)^{-1/2}\ \mathrm{Diag}(x)^{\circ r}\ \ S^{\circ r}\ (\mathrm{Diag}(x)^{\circ r})^{-1}\mathrm{Diag}(t)^{-1/2}\mathrm{Diag}(t)^{-1/2} \\
&= (\mathrm{Diag}(t)^{1/2}\mathrm{Diag}(x)^{\circ r})\ (\mathrm{Diag}(t)^{-1/2}S^{\circ r}\mathrm{Diag}(t)^{-1/2})\ (\mathrm{Diag}(t)^{1/2}\mathrm{Diag}(x)^{\circ r})^{-1}.
\end{aligned}$$

Since the matrix $\mathrm{Diag}(t)^{-1/2}S^{\circ r}\mathrm{Diag}(t)^{-1/2}$ equals $S^{\circ r} \circ T$, the lemma holds. □

COROLLARY 4.6. M z, S T 4.5

(i) $\Gamma_r M$ $r \in \mathbb{R}$

(ii) M dpsd $\Gamma_r M$ dpsd $r \in \mathbb{N}$ M dpd $\Gamma_r M$ dpd $r \in \mathbb{N}$

. Statement (i) follows immediately from Theorem 4.5. Statement (ii) follows from the fact that a Hadamard product of matrices is positive (semi-) definite if

each of the factors is positive (semi-) definite. Moreover, if at least one of the factors is positive definite, and none of the other factors has a zero diagonal entry, then the product is positive definite (see, e.g., [35, p. 309], or [23]). These are basic results in the theory of Hadamard products, an area now covered by a vast body of literature. Standard references in this area are [3, 35]. It should be noted that $r \in \mathbb{N}$ is in general a necessary condition [35, p. 453]. □

THEOREM 4.7. *M*
$(M, e_{(i)}, r_{(i)})$
(i)
(ii) $r_i = 2$ $e_i = 2$
$(M, e_{(i)}, r_{(i)})$ dpsd

These statements[1] follow from the fact that Exp_2 maps diagonally symmetric matrices onto matrices and from Corollary 4.6. □

Theorem 4.7 represents a qualitative result on the process. Under fairly basic assumptions the spectra of the iterands are real and nonnegative. In [11] it was furthermore proven that the process converges quadratically in the neighborhood of nonnegative -invariant matrices. These combined facts indicate that the process has a sound mathematical foundation. Still, much less can be said about the connection between successive iterands than in the case of the discrete Markov process.

The question now rises whether the process can be further studied aiming at quantitative results. It was seen that $\Gamma_r M$, $r \in \mathbb{N}$ can be described in terms of a Hadamard product of positive semi-definite matrices relating the symmetric matrices associated with M and $\Gamma_r M$ (in Theorem 4.5). There are many results on the spectra of such products. The results are generically in terms of a majorization relationship such as

$$\sum_{i=1}^{k} \sigma_i(A \circ B) \;\leq\; \sum_{i=1}^{k} f_i(A)\sigma_i(B), \quad k = 1, \ldots, n.$$

Here $\sigma_i()$ denotes the i-largest singular value, and $f_i(A)$ may stand (among others) for the i-largest singular value of A, the i-largest diagonal entry of A, the i-largest Euclidean column length, or the i-largest Euclidean row length. Well-known references in this field are [3, 35]. Unfortunately such inequalities go the wrong way in a sense. Since the inflation operator has apparently the ability to press several large eigenvalues towards 1, what is needed are inequalities of the type

$$\sum_{i=1}^{k} \sigma_i(A \circ B) \;\geq\; \text{(something nice here)}.$$

However, the number of eigenvalues pressed towards 1 by Γ_r can be any number including zero (noting that one eigenvalue 1 is always present). Moreover, Γ_r also has the ability to press small eigenvalues towards zero. Clearly, one cannot expect to find inequalities of the "$\geq$" type without assuming additional characteristics of M. It is shown in the next section that the classic majorization relation formulated in

[1] Clearly the condition under (ii) can be weakened; it is only necessary that e_i is at least one time even for an index $i = k$ such that $r_i \in \mathbb{N}$ for $i \geq k$. However, the assumptions under (ii) can be viewed as a standard way of enforcing convergence in a setting genuinely differing from the discrete Markov process.

Lemma 4.4 (a) between the eigenvalues and diagonal entries of a $\cdot,\cdot$ matrix, plus a classification of the diagonal entries of a $\cdot,\cdot$ matrix, gives useful information on the relationship between eigenvalues of a stochastic $\cdot,\cdot$ matrix and its image under Γ_r.

5. Structure in *dpsd* matrices. The main objective for the remainder of this paper is to establish structure theory for the class of $\cdot,\cdot$ matrices and study the behavior of Γ_∞ using these results. It will be shown that for stochastic $\cdot,\cdot$ M the spectrum of the matrix Γ_∞ is of the form $\{0^{n-k}, 1^k\}$, where k is related to a structural property of M. Throughout this section two symbols are used that are associated with a $\cdot,\cdot$ matrix A, namely the symbol $\looparrowright$ which denotes an arc relation defined on the indices of A, and the symbol $\sim$ which denotes an equivalence relation on the indices of A. It should be clear from the context which matrix they refer to. All results in this section are stated in terms of columns; the analogous statements in terms of rows hold as well.

DEFINITION 5.1. $\cdot$ A $\cdot$ dpsd $\cdot,\cdot$ $\cdot$ n $\cdot,\cdot$ $\cdot$ k $\cdot$ l $\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $1,\ldots,n$

(i) $\cdot,\cdot$ $\cdot,\cdot$ $\sim$ $\cdot,\cdot$ $\cdot,\cdot$ $\{1,\ldots,n\}$ $\cdot$ $k \sim l \equiv$ columns k and l of A are scalar multiples of each other via scalars on the complex unit circle

(ii) $\cdot,\cdot$ $\cdot,\cdot$ $\looparrowright$ $\cdot,\cdot$ $\cdot,\cdot$ $\{1,\ldots,n\}$ $\cdot,\cdot$ $p \neq q$ $\cdot$ $q \looparrowright p \equiv |A_{pq}| \geq |A_{qq}|$

(iii) $\cdot$ E $\cdot$ F $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $\{1,\ldots,n\}/\sim$ $\cdot,\cdot$ $\cdot,\cdot$ $\looparrowright$ $\cdot,\cdot$ $F \looparrowright E \equiv \exists e \in E, \exists f \in F[f \looparrowright e]$ $\cdot,\cdot$ $\looparrowright$ $\cdot,\cdot$ $\sim$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $\forall e' \in E, \forall f' \in F[f' \looparrowright e']$

LEMMA 5.2. $\cdot$ A $\cdot$ dpsd $\cdot,\cdot$ $\cdot$ n $\cdot,\cdot$ $\cdot$ k $\cdot$ l $\cdot,\cdot$ $\cdot,\cdot$ $1,\ldots,n$ $\cdot,\cdot$

$$l \looparrowright k \wedge k \looparrowright l \cdot,\cdot \cdot,\cdot k \sim l$$

This follows from Lemma 4.4 (b) and the fact that the assumption implies $\det A[kl|kl] = 0$. The following lemma prepares for a mapping of $\cdot,\cdot$ matrices onto directed acyclic graphs.

LEMMA 5.3. $\cdot$ A $\cdot$ dpsd $\cdot,\cdot$ $\cdot$ n $\cdot,\cdot$ $\cdot,\cdot$ $\cdot$ k $\cdot,\cdot$ $\cdot,\cdot$ $p_i, i = 1,\ldots,k, k > 1$ $\cdot,\cdot$ $\cdot,\cdot$ $p_1 \looparrowright p_2 \looparrowright \ldots \looparrowright p_k \looparrowright p_1$ $\cdot,\cdot$ $p_1 \sim p_2 \sim \ldots \sim p_k$ $\cdot,\cdot$ $\cdot,\cdot$ $p_i, i = 1,\ldots,k$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $\{1,\ldots,n\}/\sim$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ A $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $A[p_1 \ldots p_k|p_i]$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ $\cdot,\cdot$ k

$\cdot,\cdot$. Without loss of generality, assume $1 \looparrowright 2 \looparrowright \ldots \looparrowright k \looparrowright 1$. The following inequalities hold, where the left-hand side inequalities follow from the inequalities implied by $\det A[i\,i{+}1] \geq 0$ and $i \looparrowright i+1$,

$$\begin{array}{ccccc} |A_{i\,i+1}| & \leq & |A_{i+1\,i+1}| & \leq & |A_{i+2\,i+1}| \\ |A_{k-1\,k}| & \leq & |A_{kk}| & \leq & |A_{1k}| \\ |A_{k1}| & \leq & |A_{11}| & \leq & |A_{21}|. \end{array}$$

Now let x be positive such that $x_q A_{pq} = x_p A_{qp}$. On the one hand, $|A_{kk}| \leq |A_{1k}|$. On

the other hand,

$$\begin{aligned} |A_{kk}| &\geq |A_{k-1\,k}| \\ &= \tfrac{x_{k-1}}{x_k}|A_{k\,k-1}| \\ &\geq \tfrac{x_{k-1}}{x_k}|A_{k-2\,k-1}| \\ &= \tfrac{x_{k-1}}{x_k}\tfrac{x_{k-2}}{x_{k-1}}|A_{k-1\,k-2}| \\ &\cdots \\ &\geq \tfrac{x_{k-1}}{x_k}\tfrac{x_{k-2}}{x_{k-1}}\cdots\tfrac{x_1}{x_2}|A_{k1}| \\ &= \tfrac{x_1}{x_k}|A_{k1}| \\ &= |A_{1k}|. \end{aligned}$$

This implies that $|A_{k-1\,k}| = |A_{kk}| = |A_{1k}|$ and the identities $|A_{i-1\,i}| = |A_{ii}| = |A_{i+1\,i}|$ are established by abstracting from the index k. From this it follows that $\det A[i, i+1|i, i+1] = 0$, and consequently $i \sim i+1$ for $i = 1, \ldots, k-1$ by Lemma 5.2. The identities $|A_{i-1\,i}| = |A_{ii}| = |A_{i+1\,i}|$ also imply the last statement of the lemma. □

Lemma 5.2 can now be generalized towards Theorem 5.4.

THEOREM 5.4. [illegible] A [illegible] dpsd [illegible] n

[illegible] $\mapsto$ [illegible] [illegible] $(\ldots)$ [illegible] $\{1, \ldots, n\}/\sim$

Note that the theorem is stated in a columnwise manner. The analogous statement for rows is of course also true. The proof of this theorem follows from Lemma 5.3.

THEOREM 5.5. [illegible] M [illegible] dpsd [illegible] n [illegible] D [illegible] [illegible] $\Gamma_\infty M$ [illegible] $\{1, \ldots, n\}/\sim$ [illegible] 5.1 [illegible] 5.4 [illegible] d [illegible] D [illegible] D [illegible] k [illegible] $\{1, \ldots, n\}/\sim$ [illegible] D [illegible] $(\ldots)$ [illegible] p [illegible] M_{pp} [illegible] p

[illegible] $\Gamma_\infty M$ [illegible] $\{0^{n-k}, 1^k\}$ [illegible] $(\Gamma_\infty M)^d$ [illegible]

[illegible]. For the duration of this proof, write S_A for the symmetric matrix to which a diagonally symmetric matrix A is similar. For the first statement, consider the identity

$$S_{(\Gamma_r M)} = \left[\Gamma_r M \circ (\Gamma_r M)^T\right]^{\circ 1/2}$$

given in Lemma 4.2. The matrices $\Gamma_r M$ and $S_{\Gamma_r M}$ have the same spectrum. Now, let r approach infinity. The identity is in the limit not meaningful, since $\Gamma_\infty M$ is not necessarily diagonalizable, and thus the left-hand side may not exist in the sense that there is no symmetric matrix to which $\Gamma_\infty M$ is similar. However, the identity [spectrum of $\Gamma_\infty M$ = spectrum of $\left[\Gamma_\infty M \circ (\Gamma_\infty M)^T\right]^{\circ 1/2}$] [illegible] remain true, since the spectrum depends continuously on matrix entries [34, p. 540], and both limits exist. Thus, it is sufficient to compute the spectrum of S_∞, which is defined as

$$S_\infty = \left[\Gamma_\infty M \circ (\Gamma_\infty M)^T\right]^{\circ 1/2}.$$

Note that the nonzero entries of $\Gamma_\infty M$ correspond with the entries of M which are maximal in their column. Whenever $[\Gamma_\infty M]_{kl} \neq 0$ [illegible] $[\Gamma_\infty M]_{lk} \neq 0$, it is true that $k \looparrowright l$ and $l \looparrowright k$. Now consider a column q in S_∞, and assume that all nonzero entries in column q of S_∞ are enumerated $S_{\infty p_i q} \neq 0$, for $i = 1, \ldots, t$. It follows that $q \looparrowright p_i \wedge p_i \looparrowright q$ for all i, thus $q \sim p_i$ for all i, and $S_\infty[p_1 \ldots p_t | p_1 \ldots p_t]$ is a positive submatrix equal to $t^{-1}J_t$, where J_t denotes the all-one matrix of dimension t. This implies that S_∞ is block diagonal (after permutation), with each block corresponding with an equivalence class in $\{1, \ldots, n\}/\sim$ which has no outgoing arc in the $\looparrowright$ arc relation. Each block contributes an eigenvalue 1 to the spectrum of S_∞. Since the spectrum of S_∞ equals the spectrum of $\Gamma_\infty M$, and there are assumed to be k equivalence classes with the stated properties, this proves the first statement.

A second approach proves both the first and the second statement. Consider $\Gamma_\infty M$ and the DAG D associated with it. Each index i for which $[\Gamma_\infty M]_{ii} \neq 0$ must be in an endclass of D because Γ_∞ annihilates all but the maximal elements in each column. Moreover, the nonzero diagonal block (possibly 1-dimensional) associated with such an index is idempotent. This implies that $\Gamma_\infty M$ can be decomposed into an idempotent part (consisting of the diagonal block) and a nilpotent part (the rest). Some calculations now verify that $(\Gamma_\infty M)^d$ is idempotent, where d is the depth of D. □

Theorems 5.4 and 5.5 shed light on the structure and the spectral properties of the iterands of the [illegible] process. Theorem 5.4 also gives the means to associate an overlapping clustering with each [illegible] iterand of an [illegible] process, simply by defining the endnodes of the associated [illegible] as the unique cores of the clustering, and adding to each set of attractors all nodes that reach it.

Consider a discrete Markov process with [illegible] input matrix M. Then the difference $M^k - M^l$, $k < l$, is again [illegible] (they have the same symmetrizing diagonal matrix, and the spectrum of $M^k - M^l$ is nonnegative). From this it follows that all sequences of diagonal entries $M^{(k)}{}_{ii}$, for fixed diagonal position ii, are nonincreasing. In contrast, given a stochastic [illegible] matrix M, the Γ_r operator, $r > 1$, (in the setting of [illegible] matrices) always increases some diagonal entries (at least one). The sum of the increased diagonal entries, of which there are at least k if k is the number of endnodes of the [illegible] associated with both M and $\Gamma_r M$, is a lower bound for the sum of the k largest eigenvalues of $\Gamma_r M$ (see Lemma 4.4 (a)).

The [illegible] process converges quadratically in the neighborhood of the [illegible]-invariant stable states. Proving (near-) global convergence seems to be a difficult task. I do believe, however, that a strong result will hold, where a provision has to be made for a special class of matrices, here dubbed flip-flop matrices. A flip-flop matrix M satisfies $\Gamma_2 M = M^{1/2}$. There exists a family of positive semi-definite flip-flop matrices of the form $aI_n + (1-a)n^{-1}J_n$, $n \in \mathbb{N}$ [12]. The simplest example is found in the case $n = 3$, where substituting $a = 1/2$ in the form yields a flip-flop matrix. For such a matrix it is relatively easy to prove that a small perturbation lands it on a trajectory away from the flip-flop state (with respect to alternation of Exp_2 and Γ_2) [12]. It can be noted that flip-flop matrices and circulant matrices in general form sets that are invariant under [illegible] iterations.

CONJECTURE 1. [illegible] $(M, e_{(i)}, r_{(i)})$ [illegible] $e_i = 2, r_i = 2$ [illegible] M [illegible] dpsd [illegible]

The requirement of irreducibility is present in order to exclude matrices that are

a direct sum of smaller-dimensional matrices.

6. Implementation and scalability. A mature, implementation of the MCL algorithm is available from http://micans.org/mcl/. This implementation is used in all of the references cited in the introduction. It scales subquadratically given conditions set forth below.

A fast implementation requires that the requirement of exact computation is dropped. For any interesting class of real-life graphs scaling towards tens of thousands of nodes and beyond, exact computation requires $O(N^2)$ memory resources and $O(N^3)$ time steps, where N is the number of nodes in the input graph, reflecting the basic costs of matrix multiplication. Even for sparse graphs, , iterands will fill rapidly as interesting graphs tend to be well-connected and have only few connected components.

The key observation is that in the presence of cluster structure, columns of , iterands generally possess a very skewed (weight) distribution of entries. The majority of the stochastic mass of any column is contained in a minority of the total set of nonzero entries (of that column), as inflation keeps the leveling power of expansion (multiplication) in check. In the , process limits, the matrix columns generally are extremely skewed, with a single nonzero entry per column (equaling one). This implies that , iterands never stray very far from the skewed weight distribution just described, and it suggests a way to compute a perturbed process that is tractable. That is to simply throw away some of the smallest entries, preferably adding to only a small percentage of the column weight, and rescale the remaining entries to have sum one again. This is the setup in the implementation described here.

The implementation uses a standard sparse matrix implementation where only nonzero entries are stored in arrays representing stochastic matrix columns (known as compressed column or column-major storage). During matrix multiplication, each new column is computed separately. First, the new column is computed exactly and nothing is disregarded. Then, the smallest entries are removed in a two-stage process where first entries smaller than a fixed threshold are removed, and then entries are recovered if the threshold turns out to be too severe, or more entries are removed if the threshold turns out to be insufficiently severe. The selection and recovery of entries is efficiently done using max and min heaps. The final assembly of entries is rescaled to have sum one. The implementation tracks how much mass is kept for each column during each iteration, and extensively reports on pruning characteristics.

This procedure has not yet been subjected to numerical analysis. The task appears to be nontrivial if a relationship with the effect on process limits is to be established, due to the general difficulties in analyzing the (nonlinear) , process. However, experiments on smaller graphs (with up to thousands of nodes) that allow exact computation indicate that perturbing the process in this manner has very minor impact on the resulting clusterings. The pruning reports in the setting of protein family analysis indicate rather limited pruning of stochastic mass. Additionally, nodes requiring severe pruning can be pruned in advance from the graph to allow for a more precise computation. In this respect, data preprocessing may aid , the same way it aids approaches to other large scale computational challenges.

Typically for large graphs of several hundreds of thousands of nodes, a maximum K of inbetween $1000-2000$ entries per column is kept. Newly computed columns may contain a number of nonzero entries L amounting to tens of thousands, and selecting the largest K entries from those L using threshold pruning and selection/recovery with min/max heaps has time requirements of order $O(L \log(K))$.

7. Conclusions, further research, and related research. The process presented here appears to be both of practical and mathematical interest. A clear relationship was established between matrices, a (defined on indices column or rowwise) that can be associated with every such matrix (Theorem 5.4), and the effect of the inflation operator on column stochastic matrices (Theorem 5.5). The defined on column indices of matrices generalizes the mapping of nonnegative -invariant matrices onto overlapping clusterings, and allows the association of an overlapping clustering with each matrix. In the process, the inflation step effectively strengthens the associated structure, the expansion step may change it. Many interesting and difficult questions remain. A worthy long standing goal is to prove or disprove Conjecture 1. Two more conjectures are made after the following list of objectives.

(i) For a fixed process $(\cdot, e_{(i)}, r_{(i)})$, what can be said about the basins of attraction of the process. Are they connected?
(ii) What can be said about the union of all basins of attraction for all limits corresponding with the same overlapping clustering (i.e., differing only in the distribution of attractors)?
(iii) Can the set of limits reachable from a fixed nonnegative matrix M for all processes $(M, e_{(i)}, r_{(i)})$ be characterized? Can it be related to a structural property of M?
(iv) Given a node set $I = \{1, \ldots, n\}$ and two directed acyclic graphs D_1 and D_2 defined on I, under what conditions on D_1 and D_2 does there exist a matrix M such that the s associated with M according to Theorem 5.4, via rows and columns, respectively, equals D_1 and D_2? What if M is also required to be column stochastic?
(v) Under what conditions do the clusters in the cluster interpretation of the limit of a convergent process $(M, e_{(i)}, r_{(i)})$ correspond with connected subgraphs in the associated graph of M?
(vi) For M, in which ways can the associated with M^2 be related to the associated with M?
(vii) Is it possible to specify a subclass $\mathcal{S}$ of the stochastic matrices and a subset R' of the reals larger than $\mathbb{N}$, such that $\Gamma_r M$ is in $\mathcal{S}$ if $r \in R'$ and $M \in \mathcal{S}$?

. The following is a relaxation of (iv): Given any DAG D is there a symmetric positive semi-definite matrix S such that D is the DAG associated with S (via either columns or rows)? This is easily answered in the affirmative via a constructive and inductive argument, working backwards from sinks to sources, at each step bordering the previously obtained matrix with zeros and adding a suitably constructed rank-one matrix.

. There is no obvious nontrivial hypothesis regarding item (vi), unless such a hypothesis takes quantitative properties of M into account. This is because the breaking up of strongly connected components that can be witnessed in the process is always reversible—uncoupling can only happen in the limit. With respect to (v), I conjecture the following.

CONJECTURE 2. $\mathcal{C}$ dpsd M M

Next, consider an process $(M, e_{(i)} \stackrel{c}{=} 2, r_{(i)} \stackrel{c}{=} 2)$, with M, that converges towards an -invariant matrix L, and let G be the associated graph

of M. The observations in section 2 suggest the following conjecture. Note that a graph automorphism of G implies the existence of a permutation matrix P such that $M = PMP^T$.

CONJECTURE 3. [illegible] L [illegible] (k,l) [illegible] G [illegible] k [illegible] l [illegible] L [illegible] G [illegible]

There are several lines of research that may inspire answers to the questions posed here. However, for none of them the connection seems so strong that existing theorems can immediately be applied. The main challenge is to further develop the framework in which the interplay of Γ_r and Exp_s can be studied. Hadamard-Schur theory was discussed in section 4. Perron-Frobenius theory, graph partitioning by eigenvectors (e.g., [55] and [56]), and work regarding the second largest eigenvalue of a graph (e.g., [1] and [9]), are a natural source of inspiration, and so is the theory of Perron complementation and stochastic complementation as introduced by Meyer ([47] and [48]). There are also papers that address the topic of the structure of matrices which have the subdominant eigenvalue close to the dominant eigenvalue ([30] and [53]). It should be noted that in the former paper matrices are studied that do not have nonnegative spectrum. In the setting of [illegible] matrices, much stronger results can be expected to hold regarding the relationship between uncoupling measures and spectrum. The literature on the subject of diagonal similarity does not seem to be of immediate further use, as it is often focused on scaling problems (e.g., [17] and [33]). For the study of flip-flop equilibrium states the many results on circulant matrices are likely to be valuable, for example the monograph [10], and the work on group majorization in the setting of circulant matrices in [26]. It may also be fruitful to investigate the relationship with [illegible] and the [illegible] for positive matrices, as studied in [4, 6, 8, 27, 58].

Acknowledgments. The author wishes to thank the anonymous referees for their many detailed and useful comments that have significantly contributed to the exposition of the paper.

REFERENCES

[1] N. ALON AND V. D. MILMAN, λ_1, *Isoperimetric inequalities for graphs, and superconcentrators*, J. Combin. Theory Ser. B, 38 (1985), pp. 73–88.

[2] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Classics in Applied Mathematics 9, SIAM, 1994. Corrected and extended republication of the 1979 book.

[3] R. BHATIA, *Matrix Analysis*, Graduate Texts in Mathematics 169, Springer-Verlag, New York, 1997.

[4] G. BIRKHOFF, *Lattice Theory*, AMS Colloquium Publications 25, 3rd ed., American Mathematical Society, Providence, RI, 1967.

[5] S. BROHÉE AND J. VAN HELDEN, *Evaluation of clustering algorithms for protein-protein interaction networks*, Bioinformatics, 7 (2006) p. 488; available online at http://www.biomedcentral.com/1471-2105/7/488/abstract.

[6] P. J. BUSHELL, *Hilbert's metric and positive contraction mappings in a Banach space*, Arch. Rational Mech. Anal., 52 (1973), pp. 330–338.

[7] Y. CHEN, Y. ZHANG, Y. YIN, G. GAO, S. LI, Y. JIANG, X. GU, AND J. LUO, *Spd—a web-based secreted protein database*, Nucleic Acids Res., 33 (2005), pp. D169–D173.

[8] J. E. COHEN, *Contractive inhomogeneous products of nonnegative matrices*, Math. Proc. Cambridge Philos. Soc., 86 (1979), pp. 351–364.

[9] D. Cvetković and S. Simić, *The second largest eigenvalue of a graph (a survey)*, Filomat, 9 (1995), pp. 449–472.
[10] P. J. Davis, *Circulant Matrices*, John Wiley & Sons, New York, 1979.
[11] S. van Dongen, *A Cluster Algorithm for Graphs*, Technical report, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam, 2000.
[12] S. van Dongen, *Graph Clustering by Flow Simulation*, Ph.D. thesis, University of Utrecht, The Netherlands, 2000.
[13] B. Dorow and D. Widdows, *Discovering corpus-specific word senses*, in the Tenth Annual Conference of the European Chapter of the Association for Computational Linguistics, Conference Companion, Bergen, Norway, 2003, pp. 79–82.
[14] B. Dorow, D. Widdows, K. Ling, J.-P. Eckmann, D. Sergi, and E. Moses, *Using curvature and Markov clustering in graphs for lexical acquisition and word sense discrimination*, arXiv, (2004) available online at http://arxiv.org/pdf/cond-mat/0403693.
[15] B. Dujon et al., *Genome evolution in yeasts*, Nature, 430 (2004), pp. 35–44.
[16] Eichinger et al., *The genome of the social amoeba dictyostelium discoideum*, Nature, 435 (2005), pp. 43–57.
[17] T. Elfving, *On some methods for entropy maximization and matrix scaling*, Linear Algebra Appl., 34 (1980), pp. 321–339.
[18] A. J. Enright, S. van Dongen, and C. A. Ouzounis, *An efficient algorithm for the large-scale detection of protein families*, Nucleic Acids Res., 7 (2002), pp. 1575–1584.
[19] A. J. Enright, V. Kunin, and C. A. Ouzounis, *Protein families and tribes in genome sequence space*, Nucleic Acids Res., 31 (2003), pp. 4632–4638.
[20] B. S. Everitt, *Cluster Analysis*, 3rd ed., Hodder & Stoughton, London, 1993.
[21] J. Falkner, F. Rendl, and H. Wolkowicz, *A computational study of graph partitioning*, Math. Programming, 66 (1994), pp. 211–239.
[22] M. Fiedler, *Special matrices and their applications in numerical mathematics*, Martinus Nijhoff Publishers, Dordrecht, 1986.
[23] C. H. Fitzgerald and R. A. Horn, *On fractional Hadamard powers of positive definite matrices*, J. Math. Anal. Appl., 61 (1977), pp. 633–342.
[24] A. R. R. Forrest et al., *Phosphoregulators: Protein kinases and protein phosphatases of mouse*, Genome Research, 13 (2003), pp. 1443–1454.
[25] D. Gfeller, J.-C. Chappelier, and P. De Los Rios, *Synonym dictionary improvement through markov clustering and clustering stability*, in Proceedings of the International Symposium on Applied Stochastic Models and Data Analysis, J. Janssen and P. Lenca, eds., 2005, pp. 106–113.
[26] A. Giovagnoli and H. P. Wynn, *Cyclic majorization and smoothing operators*, Linear Algebra Appl., 239 (1996), pp. 215–225.
[27] J. Hajnal, *On products of non–negative matrices*, Math. Proc. Cambridge Philos. Soc., 79 (1976), pp. 521–530.
[28] N. Hall et al., *A comprehensive survey of the plasmodium life cycle by genomic, transcriptomic, and proteomic analyses*, Science, 307 (2005), pp. 82–86.
[29] T. J. Harlow, J. Peter Gogarten, and M. A. Ragan, *A hybrid clustering approach to recognition of protein families in* 114 *microbial genomes*, BMC Bioinformatics, 5 (2004), p. 45.
[30] D. J. Hartfiel and C. D. Meyer, *On the structure of stochastic matrices with a subdominant eigenvalue near* 1, Linear Algebra Appl., 1272 (1998), pp. 193–203.
[31] D. J. Hartfiel and J. W. Speelman, *Diagonal similarity of irreducible matrices to row stochastic matrices*, Pacific J. Math., 40 (1972), pp. 97–99.
[32] D. Heesch and S. Rüger, *NN^k networks for content-based image retrieval*, in McDonald and Tait [45], pp. 253–266.
[33] D. Hershkowitz, W. Huang, M. Neumann, and H. Schneider, *Minimization of norms and the spectral radius of a sum of nonnegative matrices under diagonal equivalence*, Linear Algebra Appl., 241/243 (1996), pp. 431–453.
[34] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1990.
[35] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.
[36] N. Jardine and R. Sibson, *Mathematical Taxonomy*, Wiley Series in Probabilistic and Mathematical Statistics, John Wiley & Sons, London, 1971.
[37] V. Kunin, I. Cases, A. J. Enright, V. de Lorenzo, and C. A. Ouzounis, *Myriads of protein families, and still counting*, Genome Biology, 5 (2003), p. 401.

[38] P. Larsson et al., *The complete genome sequence of francisella tularensis, the causative agent of tularemia*, Nature Genetics, 37 (2005), pp. 153–159.
[39] D. A. Lee et al., *Eyesite: A semi-automated database of protein families in the eye*, Nucleic Acids Res., 32 (2004), pp. D148–D152.
[40] R. Leplae, A. Hebrant, S. J. Wodak, and A. Toussaint, *Aclame: A classification of mobile genetic elements*, Nucleic Acids Res., 32 (2004), pp. D45–D49.
[41] L. Li, C. J. Stoeckert, and D. S. Roos, *Orthomcl: Identification of ortholog groups for eukaryotic genomes*, Genome Research, 13 (2003), pp. 2178–2189.
[42] R. Loewy, *Diagonal similarity of matrices*, Portugaliae Mathematica, 43 (1985–1986), pp. 55–59.
[43] L. Lovász, *Random walks on graphs: A survey*, in Miklos et al. [49], pp. 353–397.
[44] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and its Applications*, Mathematics in Science and Engineering 143, Academic Press, New York, 1979.
[45] S. McDonald and J. Tait, eds., *Advances in Information Retrieval: 26th European Conference on IR Research, ECIR* 2004, Lecture Notes in Comput. Sci. 2997, Springer-Verlag, Heidelberg, 2004.
[46] J. McPherson, K.-L. Ma, and M. Ogawa, *Discovering parametric clusters in social small-world graphs*, in The 20th Annual ACM Symposium on Applied Computing, 2005.
[47] C. D. Meyer, *Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems*, SIAM Rev., 31 (1989), pp. 240–272.
[48] C. D. Meyer, *Uncoupling the Perron eigenvector problem*, Linear Algebra Appl., 114/115 (1989), pp. 69–74.
[49] D. Miklos et al., eds., *Combinatorics, Paul Erdős is eighty*, vol. II, Janos Bolyai Mathematical Society, 1996.
[50] H. Minc, *Nonnegative Matrices*, Wiley Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, 1988.
[51] B. Mirkin, *Mathematical Classification and Clustering*, Kluwer Academic Publishers, Boston, 1996.
[52] J. Parkinson et al., *A transcriptomic analysis of the phylum nematoda*, Nature Genetics, 36 (2004), pp. 1259–1267.
[53] B. N. Parlett, *Invariant subspaces for tightly clustered eigenvalues of tridiagonals*, BIT, 36 (1996), pp. 542–562.
[54] J. B. Pereira-Leal, A. J. Enright, and C. A. Ouzounis, *Detection of functional modules from protein interaction networks*, PROTEINS: Structure, Function, and Bioinformatics, 54 (2004), pp. 49–57.
[55] A. Pothen, H. D. Simon, and K.-P. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.
[56] D. L. Powers, *Structure of a matrix according to its second eigenvector*, in Current trends in matrix theory. Proceedings of the Third Auburn Matrix Theory Conference, Auburn University, Auburn, AL, 1986, F. Uhlig and R. Grone, eds., Elsevier, 1987, pp. 261–265.
[57] B. G. L. Ramaswamy and L. Liu, *Connectivity based node clustering in decentralized peer-to-peer networks*, in the Third International Conference on Peer-to-Peer Computing (ICP2PC 2003), Linkoping, Sweden, 2003.
[58] E. Seneta, *Nonnegative Matrices and Markov Chains*, 2nd ed., Springer, Berlin, 1981.
[59] A. Sinclair, *Algorithms for Random Generation and Counting, A Markov Chain Approach*, Progress in Theoretical Computer Science, Birkhäuser Boston, Boston, 1993.
[60] L. D. Stein et al., *The genome sequence of caenorhabditis briggsae: A platform for comparative genomics*, PLoS Biology, 1 (2003), pp. 166–192.
[61] J. D. Watson et al., *Target selection and determination of function in structural genomics*, IUBMB Life, 55 (2003), pp. 249–255.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 142–153

A DIVIDE-AND-CONQUER METHOD FOR THE TAKAGI FACTORIZATION*

WEI XU† AND SANZHENG QIAO†

Abstract. This paper presents a divide-and-conquer method for computing the symmetric singular value decomposition, or Takagi factorization, of a complex symmetric and tridiagonal matrix. An analysis of accuracy shows that our method produces accurate Takagi values and orthogonal Takagi vectors. Our preliminary numerical experiments have confirmed our analysis and demonstrated that our divide-and-conquer method is much more efficient than the implicit QR method even for moderately large matrices.

Key words. divide-and-conquer method, symmetric SVD, Takagi factorization

AMS subject classifications. 15A18, 65F20, 65F25, 65F50

DOI. 10.1137/050624558

1. Introduction.

The Takagi factorization of a complex symmetric matrix A can be written as [7]

$$A = V\Sigma V^T,$$

where V is a unitary matrix, V^T is the transpose of V, and Σ is a nonnegative diagonal matrix. The columns of V are called the Takagi vectors of A and the diagonal elements of Σ are its Takagi values. Since $V^T = \bar{V}^H$, where $\bar{V}$ denotes the complex conjugate of V, the Takagi factorization is a symmetric form of the singular value decomposition (SVD); but there are differences. A pair of left-right singular vectors are unique up to a complex scaling factor with unit modulus, while the Takagi vectors are unique up to a sign change. Therefore, if $\mathbf{v}_i$ is a Takagi vector, then $(\mathbf{v}_i,\ \bar{\mathbf{v}}_i)$ is a pair of left-right singular vectors, but a left singular vector is not necessarily a Takagi vector; see an example below.

Similar to the computation of the SVD, a standard algorithm for computing the Takagi factorization consists of two stages. The first stage reduces a complex symmetric matrix A of order n to a complex symmetric tridiagonal matrix:

$$A = PTP^T \equiv P \begin{bmatrix} a_1 & b_1 & & 0 \\ b_1 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ 0 & & b_{n-1} & a_n \end{bmatrix} P^T, \tag{1}$$

where P is a unitary matrix of order n and T is tridiagonal. For example, the Lanczos tridiagonalization method with partial orthogonalization [9, 12] can be used. The second stage computes the Takagi factorization $T = Q\Sigma Q^T$ of the complex symmetric tridiagonal T. Combining the two stages, we have

$$A = P(Q\Sigma Q^T)P^T = V\Sigma V^T,$$

*Received by the editors February 17, 2005; accepted for publication (in revised form) by P. C. Hansen August 12, 2007; published electronically February 20, 2008.
http://www.siam.org/journals/simax/30-1/62455.html

†Department of Computing and Software, McMaster University, Hamilton, ON L8S 4K1, Canada (xuw5@mcmaster.ca, qiao@mcmaster.ca).

where $V = PQ$.

In this paper, we focus on the computation of the Takagi factorization of the complex symmetric tridiagonal T using the divide-and-conquer method based on rank-one tearing of TT^H. It is known that the divide-and-conquer method is one of the most efficient methods for computing the eigenvalues and eigenvectors of a large, normally of order larger than dozens, Hermitian tridiagonal matrix [3]. Apparently, the Takagi vectors of T—that is, the columns of Q—are the eigenvectors of the positive semidefinite Hermitian matrix TT^H, since $TT^H = Q\Sigma Q^T \bar{Q}\Sigma Q^H = Q\Sigma^2 Q^H$. However, an eigenvector of TT^H may not be a Takagi vector of T. For example, let

$$T = \begin{bmatrix} 1 & i \\ i & -1 \end{bmatrix}, \quad \text{where} \quad i = \sqrt{-1};$$

then

$$TT^H = \begin{bmatrix} \frac{\sqrt{2}}{4} + \frac{\sqrt{6}}{4}i & -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{6}}{4} + \frac{\sqrt{2}}{4}i & \frac{\sqrt{2}}{2}i \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{4} - \frac{\sqrt{6}}{4}i & -\frac{\sqrt{6}}{4} - \frac{\sqrt{2}}{4}i \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}i \end{bmatrix}$$

is an eigenvalue decomposition of TT^H. Using the algorithm in [10], we can obtain the Takagi factorization

$$T = Q\Sigma Q^T = \begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{1}{2} + \frac{1}{2}i \\ -\frac{\sqrt{2}}{2}i & \frac{1}{2} - \frac{1}{2}i \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}i \\ \frac{1}{2} + \frac{1}{2}i & \frac{1}{2} - \frac{1}{2}i \end{bmatrix}.$$

In fact, it is shown in [7, Corollary 4.4.5] that if A is complex symmetric and the eigenvalues of AA^H are distinct, and if $AA^H = U\Sigma^2 U^H$, then there exists a diagonal matrix $D = \mathrm{diag}(e^{i\theta_1}, \ldots, e^{i\theta_n})$ with real θ_i such that $A = V\Sigma V^T$ with $V = UD$. The diagonal entries of D are determined by the relation $A\bar{U} = U\Sigma D^2$. In the above example, $e^{i\theta_1} = -1/2 + \sqrt{3}\,i/2$ and $e^{i\theta_2} = -\sqrt{2}/2 - \sqrt{2}\,i/2$. However, if σ_i^2 is a multiple eigenvalue of AA^H, then, following the proof of Theorem 4.4.3 in [7], we can construct the Takagi vector $\mathbf{v}_i$ corresponding to the singular value σ_i of A from the eigenvector $\mathbf{u}_i$ corresponding to σ_i^2 using

$$\mathbf{v}_i = \alpha_i(A\bar{\mathbf{u}}_i + \sigma_i \mathbf{u}_i),$$

where $\alpha_i = 1/\|A\bar{\mathbf{u}}_i + \sigma_i \mathbf{u}_i\|_2$ is the normalization factor. The details of the transformation will be described in section 3.

The basic idea behind our method is to apply the divide-and-conquer method to TT^H to compute its eigenvectors and eigenvalues. The square roots of the eigenvalues of TT^H are the Takagi values of T. Since an eigenvector of TT^H may not be a Takagi vector of T, we then transform the eigenvectors of TT^H into the Takagi vectors of T. However, explicitly computing TT^H is too expensive and also destroys the tridiagonal structure of T. We will introduce an implicit method for computing the eigenvalue decomposition of TT^H.

The rest of this paper is organized as follows. Section 2 describes a divide-and-conquer method for computing the eigenvalue decomposition of TT^H without explicitly forming TT^H. In section 3, we propose a method for transforming the eigenvectors of TT^H into the Takagi vectors of T. We analyze the sensitivity of the Takagi vectors of T in section 4. Finally, our preliminary numerical experiments are demonstrated in section 5 to show the stability, accuracy, and efficiency of our algorithm.

2. Divide-and-conquer scheme. Let the Takagi factorization of the complex symmetric tridiagonal matrix T in (1) be

$$Q^H T \bar{Q} \;=\; \Sigma \;=\; \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \quad \text{or} \quad T = Q \Sigma Q^T.$$

In the first step, we tear the tridiagonal matrix T into two tridiagonal submatrices of half size. For simplicity, we assume that n is a power of 2 and $m = n/2$; then

$$T = \begin{bmatrix} T_1 & b_m \mathbf{e}_m \mathbf{e}_1^T \\ b_m \mathbf{e}_1 \mathbf{e}_m^T & T_2 \end{bmatrix}, \tag{2}$$

where

$$T_1 = \begin{bmatrix} a_1 & b_1 & & 0 \\ b_1 & \ddots & \ddots & \\ & \ddots & \ddots & b_{m-1} \\ 0 & & b_{m-1} & a_m \end{bmatrix}, \quad T_2 = \begin{bmatrix} a_{m+1} & b_{m+1} & & 0 \\ b_{m+1} & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ 0 & & b_{n-1} & a_n \end{bmatrix},$$

and $\mathbf{e}_1$ and $\mathbf{e}_m$ are unit vectors, $[1, 0, \ldots, 0]^T$ and $[0, \ldots, 0, 1]^T$, respectively. In this section, we present a divide-and-conquer method for computing the eigenvalue decomposition of TT^H given the eigenvalue decompositions of $T_1 T_1^H$ and $T_2 T_2^H$. Our method is based on the rank-one modification of symmetric eigenvalue decomposition.

2.1. Dividing the matrix. We first establish the relations between the eigenvalues and eigenvectors of $T_i T_i^H$, $i = 1, 2$, and those of TT^H as follows. From (2), we get

$$\begin{aligned}
TT^H &= \begin{bmatrix} T_1 & b_m \mathbf{e}_m \mathbf{e}_1^T \\ b_m \mathbf{e}_1 \mathbf{e}_m^T & T_2 \end{bmatrix} \begin{bmatrix} T_1^H & \bar{b}_m \mathbf{e}_m \mathbf{e}_1^T \\ \bar{b}_m \mathbf{e}_1 \mathbf{e}_m^T & T_2^H \end{bmatrix} \\
&= \begin{bmatrix} T_1 T_1^H + |b_m|^2 \mathbf{e}_m \mathbf{e}_m^T & b_m \mathbf{e}_m \mathbf{e}_1^T T_2^H + \bar{b}_m T_1 \mathbf{e}_m \mathbf{e}_1^T \\ \bar{b}_m T_2 \mathbf{e}_1 \mathbf{e}_m^T + b_m \mathbf{e}_1 \mathbf{e}_m^T T_1^H & T_2 T_2^H + |b_m|^2 \mathbf{e}_1 \mathbf{e}_1^T \end{bmatrix} \\
&= \begin{bmatrix} T_1 T_1^H & 0 \\ 0 & T_2 T_2^H \end{bmatrix} + \begin{bmatrix} |b_m|^2 \mathbf{e}_m \mathbf{e}_m^T & b_m \mathbf{e}_m \mathbf{e}_1^T T_2^H \\ \bar{b}_m T_2 \mathbf{e}_1 \mathbf{e}_m^T & 0 \end{bmatrix} \\
&\quad + \begin{bmatrix} 0 & \bar{b}_m T_1 \mathbf{e}_m \mathbf{e}_1^T \\ b_m \mathbf{e}_1 \mathbf{e}_m^T T_1^H & |b_m|^2 \mathbf{e}_1 \mathbf{e}_1^T \end{bmatrix} \\
&= \begin{bmatrix} T_1 T_1^H & 0 \\ 0 & T_2 (I_m - \mathbf{e}_1 \mathbf{e}_1^T) T_2^H \end{bmatrix} + \begin{bmatrix} b_m \mathbf{e}_m \\ T_2 \mathbf{e}_1 \end{bmatrix} \begin{bmatrix} \bar{b}_m \mathbf{e}_m^T & \mathbf{e}_1^T T_2^H \end{bmatrix} \\
&\quad + \begin{bmatrix} 0 & \bar{b}_m T_1 \mathbf{e}_m \mathbf{e}_1^T \\ b_m \mathbf{e}_1 \mathbf{e}_m^T T_1^H & |b_m|^2 \mathbf{e}_1 \mathbf{e}_1^T \end{bmatrix} \\
&= \begin{bmatrix} T_1 (I_m - \mathbf{e}_m \mathbf{e}_m^T) T_1^H & 0 \\ 0 & T_2 (I_m - \mathbf{e}_1 \mathbf{e}_1^T) T_2^H \end{bmatrix} + \begin{bmatrix} b_m \mathbf{e}_m \\ T_2 \mathbf{e}_1 \end{bmatrix} \begin{bmatrix} \bar{b}_m \mathbf{e}_m^T & \mathbf{e}_1^T T_2^H \end{bmatrix} \\
&\quad + \begin{bmatrix} T_1 \mathbf{e}_m \\ b_m \mathbf{e}_1 \end{bmatrix} \begin{bmatrix} \mathbf{e}_m^T T_1^H & \bar{b}_m \mathbf{e}_1^T \end{bmatrix} \\
&= \begin{bmatrix} T_1 (I_m - \mathbf{e}_m \mathbf{e}_m^T) T_1^H & 0 \\ 0 & T_2 (I_m - \mathbf{e}_1 \mathbf{e}_1^T) T_2^H \end{bmatrix} + \mathbf{z}_1 \mathbf{z}_1^H + \mathbf{z}_2 \mathbf{z}_2^H,
\end{aligned} \tag{3}$$

where

$$\mathbf{z}_1 = \begin{bmatrix} b_m \mathbf{e}_m \\ T_2 \mathbf{e}_1 \end{bmatrix} \quad \text{and} \quad \mathbf{z}_2 = \begin{bmatrix} T_1 \mathbf{e}_m \\ b_m \mathbf{e}_1 \end{bmatrix}.$$

From (3), if the eigenvalue decompositions

$$T_1 T_1^H = U_1 \Sigma_1^2 U_1^H \quad \text{and} \quad T_2 T_2^H = U_2 \Sigma_2^2 U_2^H \tag{4}$$

of the positive semidefinite Hermitian matrices $T_1 T_1^H$ and $T_2 T_2^H$ are available, then we can find the eigenvalue decomposition of TT^H by four rank-one modifications. Thus, if the Takagi factorizations of T_1 and T_2 are available, then we can compute the Takagi values of T and the eigenvectors of TT^H by four rank-one modifications. Later in section 3, we will show how to transform the eigenvectors into the Takagi vectors.

Now, we discuss the rank-one modification. Cuppen [2, Theorem 2.1] characterizes the eigenvalues and eigenvectors of the real symmetric rank-one modification. We generalize it to the complex case. The proof is analogous to the one in [2], so it is omitted.

THEOREM 2.1. [illegible] D^2 [illegible] $\operatorname{diag}(d_1^2, \ldots, d_n^2)$ [illegible] $d_1^2 > d_2^2 > \cdots > d_n^2$, $\mathbf{z} \in C^n$ [illegible] $\rho > 0$ [illegible] $D^2 + \rho \mathbf{z}\mathbf{z}^H$ [illegible] n [illegible] $\delta_1^2 > \delta_2^2 > \cdots > \delta_n^2$ [illegible]

$$w(\delta^2) = 1 + \rho \mathbf{z}^H (D^2 - \delta^2 I)^{-1} \mathbf{z} = 1 + \rho \sum_{j=1}^{n} \frac{|z_j|^2}{d_j^2 - \delta^2}. \tag{5}$$

[illegible] $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n$ [illegible] $D^2 + \rho \mathbf{z}\mathbf{z}^H$ [illegible]

$$\mathbf{g}_j = (D^2 - \delta_j^2 I)^{-1} \mathbf{z} / \|(D^2 - \delta_j^2 I)^{-1} \mathbf{z}\|_2, \tag{6}$$

[illegible] d_j^2 [illegible] δ_j^2

$$d_n^2 < \delta_n^2 < d_{n-1}^2 < \delta_{n-1}^2 < \cdots < d_1^2 < \delta_1^2 < d_1^2 + \rho \mathbf{z}^H \mathbf{z}.$$

Applying the above theorem, we can compute the eigenvalue decomposition of TT^H from those in (4) via four rank-one modifications. Specifically, suppose that the eigenvalue decomposition $T_1 T_1^H = U_1 \Sigma_1^2 U_1^H$ in (4) is available; then

$$\begin{aligned} &T_1 (I_m - \mathbf{e}_m \mathbf{e}_m^T) T_1^H = T_1 T_1^H - T_1 \mathbf{e}_m \mathbf{e}_m^T T_1^H \\ &= U_1 (\Sigma_1^2 - (U_1^H T_1 \mathbf{e}_m)(U_1^H T_1 \mathbf{e}_m)^H) U_1^H. \end{aligned}$$

Applying Theorem 2.1 to $-\Sigma_1^2 + (U_1^H T_1 \mathbf{e}_m)(U_1^H T_1 \mathbf{e}_m)^H$, we obtain the eigenvalue decomposition of $T_1 T_1^H - T_1 \mathbf{e}_m \mathbf{e}_m^T T_1^H$. Similarly, the eigenvalue decomposition of $T_2 T_2^H - T_2 \mathbf{e}_1 \mathbf{e}_1^T T_2^H$ can be obtained from $T_2 T_2^H = U_2 \Sigma_2^2 U_2^H$ by applying Theorem 2.1. Thus, we suppose

$$T_1 T_1^H - T_1 \mathbf{e}_m \mathbf{e}_m^T T_1^H = \hat{U}_1 \hat{\Sigma}_1^2 \hat{U}_1^H \quad \text{and} \quad T_2 T_2^H - T_2 \mathbf{e}_1 \mathbf{e}_1^T T_2^H = \hat{U}_2 \hat{\Sigma}_2^2 \hat{U}_2^H. \tag{7}$$

Applying the above decompositions to (3), we have

$$TT^H = \begin{bmatrix} \hat{U}_1 & \\ & \hat{U}_2 \end{bmatrix} \left(\begin{bmatrix} \hat{\Sigma}_1^2 & \\ & \hat{\Sigma}_2^2 \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{u}}_1 \\ \hat{\mathbf{u}}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_1 \\ \hat{\mathbf{u}}_2 \end{bmatrix}^H + \begin{bmatrix} \hat{\mathbf{v}}_1 \\ \hat{\mathbf{v}}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{v}}_1 \\ \hat{\mathbf{v}}_2 \end{bmatrix}^H \right) \begin{bmatrix} \hat{U}_1 & \\ & \hat{U}_2 \end{bmatrix}^H, \tag{8}$$

where $\hat{\mathbf{u}}_1 = b_m \hat{U}_1^H \mathbf{e}_m$, $\hat{\mathbf{u}}_2 = \hat{U}_2^H T_2 \mathbf{e}_1$, $\hat{\mathbf{v}}_1 = \hat{U_1}^H T_1 \mathbf{e}_m$, and $\hat{\mathbf{v}}_2 = b_m \hat{U}_2^H \mathbf{e}_1$. This shows that the eigenvalue decomposition of TT^H can be obtained by two more rank-one modifications.

The numerical computation of the rank-one modification, i.e., the roots of the rational function (5) and the eigenvectors (6) will be treated in section 4.

2.2. Deflation. In this subsection, we remove the assumptions of distinctiveness of the diagonal entries d_i and no zero entries in the modification vector $\mathbf{z}$ by applying the deflation technique given in [14]. We first consider the case when $\mathbf{z}$ has zero entries. It can be easily verified that $(d_i^2, \mathbf{e}_i)$ is an eigenpair of $D^2 + \rho \mathbf{z}\mathbf{z}^H$ if $z_i = 0$. In this case, the problem can be deflated by one for each zero entry in $\mathbf{z}$. Next, we consider the case when there are two equal diagonal elements in D^2, say $d_i^2 = d_j^2$. Let P be a Givens rotation such that

$$P \begin{bmatrix} z_i \\ z_j \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix};$$

then

$$P \left(\begin{bmatrix} d_i^2 & 0 \\ 0 & d_j^2 \end{bmatrix} + \begin{bmatrix} z_i \\ z_j \end{bmatrix} \begin{bmatrix} z_i \\ z_j \end{bmatrix}^H \right) P^H = \begin{bmatrix} d_i^2 & 0 \\ 0 & d_j^2 \end{bmatrix} + \begin{bmatrix} * \\ 0 \end{bmatrix} \begin{bmatrix} * \\ 0 \end{bmatrix}^H.$$

Thus, when $d_i^2 = d_j^2$ for some $i \neq j$, we can assume $z_i = 0$ or $z_j = 0$. So, the case of equal diagonal elements in D is changed to the case of zero entries in $\mathbf{z}$.

Due to the rounding errors, we regard two elements d_i^2 and d_j^2 equal if the difference between them is less than a predetermined tolerance *tol*. How do we determine the tolerance? In our deflation procedure, when d_i^2 and d_j^2 are numerically equal, we find a Givens rotation to transform $[z_i, z_j]^T$ into $[*, 0]^T$. Let $c = \bar{z}_i / \sqrt{|z_i|^2 + |z_j|^2}$ and $s = -\bar{z}_j / \sqrt{|z_i|^2 + |z_j|^2}$; then

$$\begin{bmatrix} c & -s \\ \bar{s} & \bar{c} \end{bmatrix} \begin{bmatrix} d_i^2 & \\ & d_j^2 \end{bmatrix} \begin{bmatrix} \bar{c} & s \\ -\bar{s} & c \end{bmatrix} = \begin{bmatrix} d_i^2 & \\ & d_j^2 \end{bmatrix} + E,$$

where

$$E = (d_i^2 - d_j^2) \begin{bmatrix} -|s|^2 & cs \\ \bar{c}\bar{s} & |s|^2 \end{bmatrix}.$$

We set the tolerance *tol* so that $\|E\|_F \leq \epsilon \, \|\mathrm{diag}(d_i^2, d_j^2)\|_F$ when $|d_i^2 - d_j^2| \leq tol$, where ϵ is the machine precision. Taking the Frobenius norm on E and $\mathrm{diag}(d_i^2, d_j^2)$, we get

$$\|E\|_F = \sqrt{2}\,|s|\,|d_i^2 - d_j^2| \quad \text{and} \quad \|\mathrm{diag}(d_i^2, d_j^2)\|_F \leq \sqrt{2}\, d_{\max}^2,$$

where $d_{\max} = \max(d_i, d_j)$. Thus, we set the tolerance

$$tol = \frac{d_{\max}^2}{|s|} \epsilon.$$

3. Takagi factorization. As described in the previous section, given the Takagi factorizations of T_1 and T_2 in (2), we can compute the eigenvalue decomposition $TT^H = U \Sigma^2 U^H$ through four rank-one modifications. Let $T = Q \Sigma Q^T$ be the Takagi factorization of T. It is obvious that the Takagi values of T are the square roots of

the eigenvalues of TT^H. It then remains to convert the eigenvectors of TT^H into the Takagi vectors of T. Specifically, given an eigenvector $\mathbf{u}_i$ of TT^H, we want to convert it into a vector $\mathbf{q}_i$ satisfying $T\bar{\mathbf{q}}_i = \sigma_i \mathbf{q}_i$. First, in the case when the eigenvalues are distinct, the eigenvectors of TT^H are uniquely defined up to a scaling factor with unit modulus, which implies that the Takagi vector $\mathbf{q}_i$ is a scalar multiple of the corresponding eigenvector $\mathbf{u}_i$. Let $T\bar{\mathbf{u}}_i = \xi\sigma_i\mathbf{u}_i$ for some scalar ξ such that $|\xi| = 1$, denote $\xi = e^{2i\phi}$ and define

$$\mathbf{q}_i \equiv e^{i\phi}\mathbf{u}_i. \tag{9}$$

Then

$$T\bar{\mathbf{q}}_i = e^{-i\phi}T\bar{\mathbf{u}}_i = e^{-i\phi}e^{2i\phi}\sigma_i\mathbf{u}_i = e^{i\phi}\sigma_i e^{-i\phi}\mathbf{q}_i = \sigma_i\mathbf{q}_i$$

as desired. Specifically, ξ can be obtained by $\xi = (\mathbf{u}_i^H T\bar{\mathbf{u}}_i)/\sigma_i$ if $\sigma_i \neq 0$; otherwise $\xi = 1$.

Next, in the case of multiple eigenvalues, $T\bar{\mathbf{u}}_i$ may not equal $\xi\sigma_i\mathbf{u}_i$. We construct

$$\mathbf{q}_i = \alpha_i(T\bar{\mathbf{u}}_i + \sigma_i\mathbf{u}_i), \tag{10}$$

where $\alpha_i = 1/\|T\bar{\mathbf{u}}_i + \sigma_i\mathbf{u}_i\|_2$ is the normalization factor. Then

$$T\bar{\mathbf{q}}_i = \alpha_i T(\overline{T\bar{\mathbf{u}}_i + \sigma_i\mathbf{u}_i}) = \alpha_i(T\bar{T}\mathbf{u}_i + \sigma_i T\bar{\mathbf{u}}_i) = \alpha_i(\sigma_i^2\mathbf{u}_i + \sigma_i T\bar{\mathbf{u}}_i) = \sigma_i\mathbf{q}_i.$$

Finally, we check the orthogonality of the Takagi vectors of T converted from the eigenvectors of TT^H. It is obvious that the orthogonality is maintained among the Takagi vectors corresponding to distinct Takagi values because of the orthogonality of the eigenvectors corresponding to distinct eigenvalues. Now, assume that $\mathbf{q}_i, \ldots, \mathbf{q}_{i+k-1}$ are the Takagi vectors corresponding to a multiple Takagi value σ_i of multiplicity $k > 1$. The construction of $\mathbf{q}_i$ shows that the subspace spanned by $\mathbf{q}_i, \ldots, \mathbf{q}_{i+k-1}$ is the same as the one spanned by $\mathbf{u}_i, \ldots, \mathbf{u}_{i+k-1}$, since $\mathbf{q}_i, \ldots, \mathbf{q}_{i+k-1}$ are the eigenvectors associated with σ_i^2. Thus, $\mathbf{q}_{i+t}$ $(t = 0, \ldots, k-1)$ are orthogonal to $\mathbf{q}_j$, the Takagi vector corresponding to σ_j, if $\sigma_j \neq \sigma_i$. However, the Takagi vectors corresponding to the equal Takagi values may lose their orthogonality. So, the modified Gram–Schmidt orthogonalization is applied to these vectors to restore the orthogonality. Suppose that $\mathbf{q}_{i+t}$ is one of the Takagi vectors corresponding to σ_i computed from (10), then we orthogonalize it against the previous t vectors $\mathbf{q}_i, \ldots, \mathbf{q}_{i+t-1}$ using the modified Gram–Schmidt method.

Now, we give the divide-and-conquer algorithm for computing the Takagi factorization of a complex symmetric tridiagonal matrix.

ALGORITHM 3.1. [illegible] T [illegible] $T = Q\Sigma Q^T$ [illegible] $TT^H = U\Sigma^2 U^H$ [illegible] $\mathbf{q}_i$ [illegible] T [illegible] $\mathbf{u}_i$ [illegible] TT^H

1. Partition T as (2). If T_1 and T_2 are small enough, then directly compute the eigenvalue decompositions

$$T_1T_1^H = U_1\Sigma_1U_1^H \quad \text{and} \quad T_2T_2^H = U_2\Sigma_2U_2^H.$$

   If T_1 and T_2 are large, apply this algorithm to T_1 and T_2.
2. Apply the deflation and the rank-one modification Theorem 2.1 to $T_1T_1^H - T_1\mathbf{e}_m\mathbf{e}_m^TT_1^H$ and $T_2T_2^H - T_2\mathbf{e}_1\mathbf{e}_1^TT_2^H$ to obtain their eigenvalue decompositions (7). Thus, TT^H has the form (8).

3. Compute the eigenvalue decomposition of TT^H via two rank-one modifications using the deflation and Theorem 2.1.
4. The Takagi values of T are the square roots of the eigenvalues of TT^H.
5. For a single Takagi value, its corresponding Takagi vector $\mathbf{q}_i$ is computed using (9); for a multiple Takagi value, its Takagi vector $\mathbf{q}_i$ is computed using (10) and then orthogonalized against the previously computed Takagi vectors corresponding to the same Takagi value by the modified Gram–Schmidt orthogonalization.

Finally, we present a complexity comparison between the divide-and-conquer method and the implicit QR method. Let $t(n)$ be the number of flops required by the divide-and-conquer method, then

$$\begin{aligned} t(n) &= 2t(n/2) && \text{for the two small submatrices } T_1 \text{ and } T_2 \\ &\quad + O(n^2) && \text{find the eignevalues and eigenvectors of } D+\rho\mathbf{z}\mathbf{z}^H \\ &\quad + 2.25cn^3 && \text{update } U. \end{aligned}$$

Thus, updating U is the major cost in our divide-and-conquer method. Ignoring the $O(n^2)$ terms, we get $t(n) \approx 3cn^3$. The constant c represents the deflation and is much smaller than one in practice [3]. In comparison, the implicit QR method in [10] requires about $6n^3$ flops. Hence, our divide-and-conquer method is more efficient than the implicit QR method.

4. Orthogonality of Takagi vectors. In the previous section, we presented a divide-and-conquer algorithm for computing the Takagi factorization of T. It is based on the rank-one update of the symmetric eigenvalue decomposition. Due to the rounding errors, the orthogonality of the eigenvectors computed by Theorem 2.1 may be lost. In this section, we present an analysis of the orthogonality of the computed eigenvectors and propose techniques for assuring good orthogonality. For simplicity, we assume that the given matrix in the rank-one modification is already deflated.

First, we derive a formula for the eigenvectors $\mathbf{g}_j$ in Theorem 2.1. Differentiating both sides of the function $w(t)$ in (5) with respect to t, we get

$$\|(D^2-\delta^2 I)^{-1}\mathbf{z}\|_2^2 = \sum_{j=1}^{n} \frac{|z_j|^2}{(d_j^2-\delta^2)^2} = \rho^{-1}|w'(\delta^2)|.$$

Then (6) can be rewritten as

$$\mathbf{g}_j = \left[\frac{z_1}{d_1^2-\delta_j^2}, \frac{z_2}{d_2^2-\delta_j^2}, \ldots, \frac{z_n}{d_n^2-\delta_j^2}\right] \frac{\sqrt{\rho}}{\sqrt{w'(\delta_j^2)}}. \tag{11}$$

Let $\hat{\delta}_i^2$ be a computed root of w in (5). In the following, by extending the results in [8], we show that if the relative error in $d_j^2-\hat{\delta}_i^2$ is small for all i and j, then the computed eigenvectors $\mathbf{g}_i$ have good orthogonality.

THEOREM 4.1. [illegible] $\hat{\delta}_i^2$ [illegible] $\hat{\delta}_k^2$ [illegible] w [illegible] (5) [illegible] $d_j^2-\hat{\delta}_i^2$ [illegible] $d_j^2-\hat{\delta}_k^2$ [illegible] θ_i [illegible] θ_k [illegible]

$$d_j^2-\hat{\delta}_i^2 = (d_j^2-\delta_i^2)(1+\theta_i) \quad \text{and} \quad d_j^2-\hat{\delta}_k^2 = (d_j^2-\delta_k^2)(1+\theta_k),$$

[illegible] $|\theta_i|,\ |\theta_k| \le \tau \ll 1$ [illegible] j [illegible]

$$|\hat{\mathbf{g}}_i^H \hat{\mathbf{g}}_k| = |\mathbf{g}_i^H E \mathbf{g}_k| \le \tau(2+\tau)\left(\frac{1+\tau}{1-\tau}\right)^2,$$

$\hat{\mathbf{g}}_i$ $\hat{\mathbf{g}}_k$ (11) E

(12)
$$E_{ii} = \frac{\theta_i + \theta_k + \theta_i\theta_k}{(1+\theta_i)(1+\theta_k)} \left(\frac{w'(\delta_i^2)w'(\delta_k^2)}{w'(\hat{\delta}_i^2)w'(\hat{\delta}_k^2)} \right)^{1/2}.$$

. From (11), we have

$$\begin{aligned}
&-\hat{\mathbf{g}}_i^H \hat{\mathbf{g}}_k \\
&= -\left(\sum_{j=1}^{n} \frac{|z_j|^2}{(d_j^2 - \delta_k^2)(d_j^2 - \delta_i^2)(1+\theta_i)(1+\theta_k)} \right) \frac{\rho}{(w'(\hat{\delta}_i^2)w'(\hat{\delta}_k^2))^{1/2}} \\
&= \left(\sum_{j=1}^{n} \frac{|z_j|^2}{(d_j^2 - \delta_k^2)(d_j^2 - \delta_i^2)} - \sum_{j=1}^{n} \frac{|z_j|^2}{(d_j^2 - \delta_k^2)(d_j^2 - \delta_i^2)(1+\theta_i)(1+\theta_k)} \right) \\
&\quad \frac{\rho}{(w'(\hat{\delta}_i^2)w'(\hat{\delta}_k^2))^{1/2}}
\end{aligned}$$

since $\mathbf{g}_i^H \mathbf{g}_k = 0$. Thus, we have

$$\begin{aligned}
&|\hat{\mathbf{g}}_i^H \hat{\mathbf{g}}_k| \\
&= \left| \sum_{j=1}^{n} \left(\frac{|z_j|^2}{(d_j^2 - \delta_k^2)(d_j^2 - \delta_i^2)} \right) \left(1 - \frac{1}{(1+\theta_i)(1+\theta_k)} \right) \frac{\rho}{(w'(\hat{\delta}_i^2)w'(\hat{\delta}_k^2))^{1/2}} \right| \\
&= \left| \sum_{j=1}^{n} \left(\frac{|z_j|^2}{(d_j^2 - \delta_k^2)(d_j^2 - \delta_i^2)} \right) \left(\frac{\theta_i + \theta_k + \theta_i\theta_k}{(1+\theta_i)(1+\theta_k)} \right) \left(\frac{w'(\delta_i^2)w'(\delta_k^2)}{w'(\hat{\delta}_i^2)w'(\hat{\delta}_k^2)} \right)^{1/2} \right. \\
&\quad \left. \frac{\rho}{(w'(\delta_i^2)w'(\delta_k^2))^{1/2}} \right| \\
&= |\mathbf{g}_i^H E \mathbf{g}_k| \le \|E\|_2,
\end{aligned}$$

where E is a diagonal matrix, whose diagonal elements are given by (12).

On the other hand, it is easy to show that

(13)
$$\frac{w'(\delta_i^2)}{w'(\hat{\delta}_i^2)} = \frac{\sum_{j=1}^{n} \frac{|z_j|^2}{(d_j^2 - \delta_i^2)^2}}{\sum_{j=1}^{n} \frac{|z_j|^2}{(d_j^2 - \delta_i^2)^2(1+\theta_i)^2}} \le (1+\tau)^2.$$

Substituting $w'(\delta_i^2)/w'(\hat{\delta}_i^2)$ in (12) with (13), we have

$$\max(|E_{ii}|) \le \frac{\tau + \tau + \tau^2}{(1-\tau)^2}(1+\tau)^2 = \tau(2+\tau)\left(\frac{1+\tau}{1-\tau} \right)^2.$$

This completes the proof. □

Apparently, if the roots δ_i^2 of w are computed in high accuracy, then the relative errors in $d_j^2 - \hat{\delta}_i^2$ are small, provided that the eigenvalues δ_i^2 are not clustered. Consequently, from the above theorem, the computed eigenvectors $\hat{\mathbf{g}}_i$ have good orthogonality.

We adopt the stable method in [5] for computing the roots δ_i of $w(\delta^2)$ in (5). It is well known that if two quantities x and y are close, then in finite-precision arithmetic it is more accurate to compute $x^2 - y^2$ via the formula $(x+y)(x-y)$ [6]. To avoid explicitly calculating the differences between squared quantities, we reformulate $w(\delta^2)$ in (5) as

$$w(\delta^2) = 1 + \psi_i(\mu) + \varphi_i(\mu) \equiv f_i(\mu),$$

where

$$\psi_1(\mu) = 0, \quad \varphi_1(\mu) = \sum_{j=1}^{n} \frac{|z_j|^2}{(\zeta_j - \mu)(d_j + d_i + \rho\mu)},$$

and

$$\psi_i(\mu) = \sum_{j=1}^{i-1} \frac{|z_j|^2}{(\zeta_j - \mu)(d_j + d_{i-1} + \rho\mu)}, \quad \varphi_i(\mu) = \sum_{j=i}^{n} \frac{|z_j|^2}{(\zeta_j - \mu)(d_j + d_{i-1} + \rho\mu)},$$

for $i > 1$, and

$$\zeta_j = (d_j - d_i)/\rho, \qquad \mu = (\delta - d_i)/\rho, \qquad \text{when } \delta^2 \in (d_i^2,\ (d_{i-1}^2 + d_i^2)/2),$$
$$\zeta_j = (d_j - d_{i-1})/\rho, \quad \mu = (\delta - d_{i-1})/\rho, \quad \text{when } \delta^2 \in [(d_{i-1}^2 + d_i^2)/2,\ d_{i-1}^2).$$

In the above formulation, an important property of $f_i(\mu)$ is that it can be evaluated accurately. Moreover, we have formulated the functions $\psi_i(\mu)$ and $\varphi_i(\mu)$ so that explicit calculation of the differences of squares such as $d_j^2 - d_i^2$ and $\delta^2 - d_i^2$ are avoided. There are many zero finding methods, for example, the rational interpolation [1] and bisection and its variations [11, 13]. Following [5], our algorithm for finding the zeros of $f_i(\mu)$ is based on the rational interpolation strategy [1] and its LAPACK implementation `slasd4`. Thus, from [5], the computed eigenvalues have high relative accuracy. The eigenvectors are computed from the computed eigenvalues following the method for computing the eigenvectors in [5], which guarantees numerical orthogonality. Thus, the computed Takagi vectors are numerically orthogonal since they are obtained by converting the eigenvectors.

Finding a root of $f_i(\mu)$ is an iterative process. The stopping criterion plays an important role in the accuracy of the computed roots. Similar to [4], we propose the stopping criterion:

$$|f_i(\mu)| \le \epsilon n(|\psi_i(\mu)| + |\varphi_i(\mu)| + 1). \tag{14}$$

In the following, we show that by using this criterion, the computed roots $\hat{\delta}_i^2$ of $w(\delta^2)$ are accurate.

Since $w(\delta_i^2) = 0$, we have

$$\begin{aligned} w(\hat{\delta}_i^2) &= w(\hat{\delta}_i^2) - w(\delta_i^2) = \rho \sum_{j=1}^{n} \frac{|z_j|^2}{d_j^2 - \hat{\delta}_i^2} - \rho \sum_{j=1}^{n} \frac{|z_j|^2}{d_j^2 - \delta_i^2} \\ &= \rho(\hat{\delta}_i^2 - \delta_i^2) \sum_{j=1}^{n} \frac{|z_j|^2}{(d_j^2 - \hat{\delta}_i^2)(d_j^2 - \delta_i^2)}. \end{aligned}$$

According to the stopping criterion (14), since $f_i(\mu)$ can be evaluated accurately, we have

$$|w(\hat{\delta}_i^2)| \le \epsilon n \left(1 + \rho \sum_{j=1}^{n} \frac{|z_j|^2}{|d_j^2 - \hat{\delta}_i^2|}\right) \le \rho\epsilon n \left(\sum_{j=1}^{n} \frac{|z_j|^2}{|d_j^2 - \hat{\delta}_i^2|} + \sum_{j=1}^{n} \frac{|z_j|^2}{|d_j^2 - \delta_i^2|}\right)$$

since $1 = -\rho \sum_{j=1}^{n} \frac{|z_j|^2}{d_j^2 - \delta_i^2}$. Without loss of generality, we assume δ_i^2 and $\hat{\delta}_i^2$ are in the same interval, say $(d_i^2,\ d_{i-1}^2)$. It follows that $(d_j^2 - \delta_i^2)(d_j^2 - \hat{\delta}_i^2) > 0$. So,

$$\begin{aligned} |w(\hat{\delta}_i^2)| = \rho|\hat{\delta}_i^2 - \delta_i^2| \sum_{j=1}^{n} \frac{|z_j|^2}{|(d_j^2 - \hat{\delta}_i^2)(d_j^2 - \delta_i^2)|} &\le \rho\epsilon n \left(\sum_{j=1}^{n} \frac{|z_j|^2}{|d_j^2 - \hat{\delta}_i^2|} + \sum_{j=1}^{n} \frac{|z_j|^2}{|d_j^2 - \delta_i^2|}\right) \\ &\le \rho\epsilon n(4\|D^2 + \rho\mathbf{z}\mathbf{z}^H\|_2 + |\hat{\delta}_i^2 - \delta_i^2|) \sum_{j=1}^{n} \frac{|z_j|^2}{|(d_j^2 - \hat{\delta}_i^2)(d_j^2 - \delta_i^2)|}, \end{aligned}$$

since $|d_j^2 - \hat{\delta}_i^2| + |d_j^2 - \delta_i^2| \le 2|d_j^2 - \delta_i^2| + |\hat{\delta}_i^2 - \delta_i^2| \le 4\|D^2 + \rho\mathbf{z}\mathbf{z}^H\|_2 + |\hat{\delta}_i^2 - \delta_i^2|$. From the above equation, we can get the upper bound for $|\hat{\delta}_i^2 - \delta_i^2|$:

$$|\hat{\delta}_i^2 - \delta_i^2| \le \frac{4\epsilon n\|D^2 + \rho\mathbf{z}\mathbf{z}^H\|_2}{1 - \epsilon n}.$$

In conclusion, we apply the rational interpolation zero finding method to $f_i(\mu)$ using the stopping criterion (14). We can then obtain accurate eigenvalues δ_i^2. Provided that the eigenvalues are not clustered, it results in the high relative accuracy of the difference $d_i^2 - \hat{\delta}_i^2$, which implies good orthogonality of the computed eigenvectors of TT^H.

5. Numerical examples. We programmed our divide-and-conquer Algorithm 3.1 in MATLAB and tested it on three types of complex symmetric and tridiagonal matrices. Our experiments were carried out on a server with two 2.4 GHz Xeon CPUs, 1GB RAM, and 80GB disk. The complex symmetric and tridiagonal matrices with predetermined Takagi values were generated as follows. First, a random vector uniformly distributed on $(0, 1]$ was generated and sorted in descending order as a Takagi value vector d. Then, a random unitary matrix was generated as a Takagi vector matrix V. The product $A = V\Sigma V^T$, where $\Sigma = \mathrm{diag}(d)$, was computed as a complex symmetric matrix. Finally, a complex symmetric and tridiagonal T was obtained by applying the Householder transformations to both sides of A. Denoting $\hat{Q}$ and $\hat{d}$ as the computed Takagi vector matrix and Takagi value vector, respectively, the error in the computed Takagi factorization was measured by

$$\gamma_t = \|\hat{Q}\hat{\Sigma}\hat{Q}^T - T\|_2, \quad \text{where} \quad \hat{\Sigma} = \mathrm{diag}(\hat{d}).$$

The error in the computed Takagi values was measured by

$$\gamma_v = \|d - \hat{d}\|_2,$$

and the orthogonality of the computed Takagi vector matrix $\hat{Q}$ was measured by

$$\gamma_o = \|\hat{Q}\hat{Q}^H - I\|_2.$$

TABLE 1
The Takagi factorization of five 256×256 *testing matrices with distinct Takagi values.*

Example	γ_o	γ_v	γ_t
1	1.3558E−14	3.1347E−14	4.1149E−12
2	2.1679E−14	1.0854E−14	4.3920E−12
3	9.7087E−14	8.4093E−15	1.1309E−12
4	1.1040E−14	1.2622E−14	5.5019E−12
5	3.0840E−14	1.1658E−14	1.1243E−12

1. Five random complex symmetric and tridiagonal matrices of order 256 were generated as described above. In this example, the Takagi values of each matrix were distinct. Table 1 shows that the computed Takagi values and Takagi vectors are accurate.

2. Five random complex symmetric and tridiagonal matrices of order 256 were generated. In this example, we set the five largest Takagi values equal and the four smallest Takagi values equally. Table 2 shows the results.

TABLE 2
The Takagi factorization of five 256×256 *testing matrices with multiple Takagi values of small multiplicity.*

Example	γ_o	γ_v	γ_t
1	7.5222E−12	1.1331E−14	1.0564E−12
2	2.5397E−12	1.9208E−14	2.6242E−12
3	2.4214E−12	6.0150E−14	6.1179E−12
4	1.9582E−12	4.8421E−14	3.2142E−12
5	6.3841E−12	1.0580E−14	2.4453E−12

TABLE 3
The Takagi factorization of five 256×256 *testing matrices with multiple Takagi values of large multiplicity.*

Example	γ_o	γ_v	γ_t
1	7.8816E−13	8.8186E−14	4.0040E−12
2	3.7709E−12	2.4154E−14	8.4231E−12
3	4.3532E−13	1.3427E−14	3.4808E−12
4	6.2713E−12	7.4803E−14	1.7887E−12
5	4.5237E−12	5.1166E−14	6.4702E−12

3. Five random T of order 256 were generated. In this example, however, we set the 31 largest Takagi values equal. Table 3 shows that the computed results are accurate.

For performance, we tested our algorithm on random complex symmetric and tridiagonal matrices of five different sizes. For each size, we generated five matrices and ran our divide-and-conquer (DAC) method and the implicit QR (IQR) method [10]. In our divide-and-conquer method, when the size of the submatrices T_i, for $i = 1, 2$, in (2) is less than or equal to 10, its Takagi factorization is computed directly by the implicit QR method. Table 4 shows the average running time and the average factorization error γ_t of the five matrices of same size. The results in Table 4 demonstrate that our method is significantly more efficient than the implicit QR method even for matrices of moderately large size.

TABLE 4
The performance and accuracy comparison of the divide-and-conquer (DAC) method and the implicit QR (IQR) method.

	Running time (sec)		γ_t	
matrix size	DAC method	IQR method	DAC method	IQR method
100	1.14	1.16	1.3352E−14	2.4668E−14
200	3.01	5.47	2.0272E−12	2.9772E−14
400	9.51	26.05	1.7014E−12	6.4860E−14
800	46.88	187.05	1.1338E−11	9.0250E−14
1600	286.14	2091.12	4.2198E−11	2.1552E−13

6. Conclusion. We have proposed a divide-and-conquer method for computing the Takagi factorization of a complex symmetric and tridiagonal matrix and presented an analysis, which shows that our method computes accurate Takagi values and vectors provided that the Takagi values are not clustered. Our preliminary experiments have demonstrated that our method produces accurate results even for matrices with multiple Takagi values and is much more efficient than the implicit QR method [10].

REFERENCES

[1] J. R. BUNCH, C. P. NIELSEN, AND D. C. SORENSEN, *Rank-one modification of the symmetric eigenproblems*, Numer. Math., 31 (1978), pp. 31–48.
[2] J. J. M. CUPPEN, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.
[3] J. W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[4] M. GU AND S. C. EISENSTAT, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 172–191.
[5] M. GU AND S. C. EISENSTAT, *A divide-and-conquer algorithm for the bidiagonal SVD*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 79–92.
[6] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[7] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1990.
[8] E. R. JESSUP AND D. C. SORENSEN, *A parallel algorithm for computing the singular value decomposition of a matrix*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 530–548.
[9] S. QIAO, G. LIU, AND W. XU, *Block Lanczos tridiagonalization of complex symmetric matrices*, in Advanced Signal Processing Algorithms, Architectures, and Implementations XV, F. T. Luk, ed., in Proceedings of the SPIE, Vol. 5910, 2005, pp. 285–295.
[10] F. T. LUK AND S. QIAO, *A fast singular value algorithm for Hankel matrices*, Fast Algorithms for Structured Matrices: Theory and Applications. Contemp. Math. 323, V. Olshevsky, ed., Amer. Math. Soc., 2003, pp. 169–177.
[11] D. P. O'LEARY AND G. W. STEWART, *Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices*, J. Comput. Phys., 90 (1990), pp. 497–505.
[12] S. QIAO, *Orthogonalization techniques for the Lanczos tridiagonalization of complex symmetric matrices*, Advanced Signal Processing Algorithms, Architectures, and Implementations XIV, F. T. Luk, ed., in Proceedings of the SPIE Vol. 5559, 2004, pp. 423–434.
[13] W. E. SHREVE AND M. R. STABNOW, *An eigenvalue algorithm for symmetric bordered diagonal matrices*, Current Trends in Matrix Theory, F. Uhling and R. Grone, eds., Elsevier, New York, 1987, pp. 339–346.
[14] G. W. STEWART, *Matrix Algorithm, Volume* II, Eigensystems, SIAM, Philadelphia, 2001.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 154–172

ON THE ESTIMATION OF THE DISTANCE TO UNCONTROLLABILITY FOR HIGHER ORDER SYSTEMS*

EMRE MENGI†

Abstract. A higher order dynamical system of order k is called controllable if the trajectory of the system as well as its first $k-1$ derivatives can be adjusted to pass through any given point at a finite time by choosing the input appropriately. The distance to uncontrollability is the norm of the smallest perturbation yielding an uncontrollable system. We derive a singular value minimization characterization for the distance to uncontrollability and present a trisection algorithm exploiting the singular value characterization. The algorithm is devised for low accuracy and depends on the extraction of the imaginary eigenvalues of even-odd matrix polynomials of degree $2k$ and size $2n$ with n denoting the size of the system. The well-studied first order distance to uncontrollability can be recovered as a special case.

Key words. matrix polynomials, dynamical systems, distance to uncontrollability, even-odd polynomials

AMS subject classifications. 65F15, 65K05, 93B05, 93B18

DOI. 10.1137/060658588

1. Introduction.

A fundamental question concerning the kth order continuous time-invariant dynamical system

$$K_k x^{(k)}(t) + \cdots + K_1 x'(t) + K_0 x(t) = Bu(t), \quad x(0) = x'(0) = \cdots = x^{(k-1)} = 0 \tag{1.1}$$

is the dimension of the subspace of reachable configurations at a given time t' where $B \in \mathbb{C}^{n\times m}$, $K_0, K_1, \ldots K_k \in \mathbb{C}^{n\times n}$, $x(t) \in \mathbb{C}^n$, and $u(t) \in \mathbb{C}^m$. Here $x(t)$ denotes the state vector, $u(t)$ denotes the control input, and $c_0, c_1, \ldots, c_{k-1} \in \mathbb{C}^n$ are given initial conditions. By a configuration at time t' we mean the vector consisting of $x(t')$ as well as its first $k-1$ time derivatives at time t'. We define the space of reachable configurations at time t' as

$$\mathcal{R}_{t'} = \{[\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{k-1}] \ : \quad \exists u(t) \text{ such that (1.1) satisfies } \varepsilon_0 = x(t'),\ \varepsilon_1 = x'(t'), \ldots,\ \varepsilon_{k-1} = x^{(k-1)}(t')\}.$$

We have full control over the system (1.1) if all of the configurations can be attained by choosing $u(t)$ appropriately, that is

$$\dim(\mathcal{R}_{t'}) = nk. \tag{1.2}$$

In this case the system (1.1) is called controllable. Otherwise, the system is called uncontrollable. For convenience we will frequently refer to the tuple of matrices $(K_k, \ldots, K_1, K_0, B)$ as controllable whenever the system (1.1) is controllable.

Controllability of a first order system, specifically with $k = 1$, $K_1 = I$ (the identity matrix) and $K_0 = -A$ (an arbitrary matrix), is well known [9] to be equivalent to

*Received by the editors May 1, 2006; accepted for publication (in revised form) by D. Boley, August 13, 2007; published electronically February 20, 2008. This work was supported in part by the National Science Foundation grant DMS-0412049.
http://www.siam.org/journals/simax/30-1/65858.html
†Department of Mathematics, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093 (emengi@math.ucsd.edu).

either of the conditions

$$\operatorname{rank}([B \;\; AB \;\; A^2B \;\; \cdots \;\; A^{n-1}B]) = n$$

or

$$\operatorname{rank}([A - \lambda I \;\; B]) = n \text{ for all } \lambda \in \mathbb{C}. \tag{1.3}$$

A similar characterization for the controllability of a descriptor system with $k = 1$, $K_1 = E$, and $K_0 = -A$ exists [7, 8]. In particular when E is nonsingular the controllability reduces to the condition

$$\operatorname{rank}([A - \lambda E \;\; B]) = n \text{ for all } \lambda \in \mathbb{C}. \tag{1.4}$$

When E is singular, the above condition needs to be accompanied by an additional rank condition that involves the null space of E. Throughout this paper we will assume that the leading coefficient is nonsingular and additionally, when perturbations to the leading coefficient are allowed, the leading coefficient remains nonsingular under all perturbations under consideration. (This condition is stated formally in Lemma 2.2.) Under this nonsingularity assumption, the rank characterizations (1.3) and (1.4) can be generalized to the higher order system and the nearby systems as follows. First observe that (1.1) can be embedded into the first order system

$$\tilde{x}'(t) = \mathcal{A}\tilde{x}(t) + \mathcal{B}u(t), \qquad \tilde{x}(0) = \begin{bmatrix} c_{k-1} \\ c_{k-2} \\ \vdots \\ c_0 \end{bmatrix}, \tag{1.5}$$

where

$$\tilde{x}(t) = \begin{bmatrix} x^{(k-1)}(t) \\ x^{(k-2)}(t) \\ x^{(k-3)}(t) \\ \vdots \\ x(t) \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} K_k^{-1}B \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad \text{and}$$

$$\mathcal{A} = \begin{bmatrix} -K_k^{-1}K_{k-1} & -K_k^{-1}K_{k-2} & \dots & -K_k^{-1}K_1 & -K_k^{-1}K_0 \\ I & 0 & & 0 & 0 \\ 0 & I & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & I & 0 \end{bmatrix}.$$

Now the higher order system is controllable if and only if the matrix $[\mathcal{A} - \lambda I \;\; \mathcal{B}]$ has full rank for all λ. Furthermore, for a given λ suppose

$$[\mathcal{A} - \lambda I \;\; \mathcal{B}] \begin{bmatrix} x_{k-1} \\ \vdots \\ x_0 \\ y_0 \end{bmatrix} = 0.$$

Using the definitions of $\mathcal{A}$ and $\mathcal{B}$, it is straightforward to deduce that $x_j = \lambda^j x_0$ and

$$[P(\lambda) \;\; B] \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = 0,$$

where

$$P(\lambda) = \sum_{j=0}^{k} \lambda^j K_j. \tag{1.6}$$

Therefore the null spaces of $[\mathcal{A} - \lambda I \;\; \mathcal{B}]$ and $[P(\lambda) \;\; B]$ have the same dimension, say $l \geq m$, which means $\text{rank}([\mathcal{A} - \lambda I \;\; \mathcal{B}]) = nk + m - l$ and $\text{rank}([P(\lambda) \;\; B]) = n + m - l$. We conclude that the controllability of the higher order system is equivalent to

$$\text{rank}([P(\lambda) \;\; B]) = n \text{ for all } \lambda \in \mathbb{C} \tag{1.7}$$

which was already mentioned in [18] without derivation.

Controllability is thus a rank determination problem, which cannot be performed reliably in the presence of rounding errors. A controllable system may still have nearby uncontrollable systems which potentially is an indicator of a problem with the model. Therefore in [22] for the first order system the distance to uncontrollability was defined as

$$\tau(A, B) = \inf\{\|[\Delta A \;\; \Delta B]\| : \text{the pair } (A + \Delta A, B + \Delta B) \text{ is uncontrollable}\} \tag{1.8}$$

with $\|\cdot\|$ denoting either the spectral norm or the Frobenius norm. Later Eising [10] proved that, in both cases, the distance to uncontrollability is equivalent to a minimization problem involving complex vectors of size n

$$\tau(A, B) = \inf_{q \in \mathbb{C}^n, \|q\| = 1} \sqrt{q^* B B^* q + q^* A (I - qq^*) A^* q} \tag{1.9}$$

and a singular value minimization problem, i.e.,

$$\tau(A, B) = \inf_{\lambda \in \mathbb{C}} \sigma_{\min}([A - \lambda I \;\; B]), \tag{1.10}$$

where $\sigma_{\min}$ denotes the smallest singular value. The most efficient computational techniques for the distance to uncontrollability exploit the definition (1.10), though there are hybrid-algorithms [24] developed following Eising's characterizations that make use of both (1.9) and (1.10). Boley observed the connection between the sensitivity of the Kronecker structure of a matrix pencil and distance to uncontrollability and based on (1.10) suggested a practical but an imprecise way to approximate the distance by solving a standard eigenvalue problem [1]. Byers introduced classes of algorithms working on one dimensional or two dimensional grids [5] to minimize $\sigma_{\min}([A - \lambda I \;\; B])$. Later Gao and Neumann [11] and He [16] modified Byers' idea for more efficient computation. Byers' grid-based algorithms and its successors are well-suited for the computation of the distance to uncontrollability with a few digits of precision but are too costly for high accuracy. Gu's bisection algorithm [14] is the first technique that retrieves the global minimum for the problem (1.10) within a factor of two without depending on a grid. Gu's algorithm later was improved by Burke, Lewis, and Overton [3] who suggested a trisection algorithm that computes $\tau(A, B)$ to arbitrary precision. With $O(n^6)$ complexity[1] these algorithms are applicable only to small systems. In [15], it is described how we can benefit from inverse iteration and shift-and-invert preconditioned Arnoldi to reduce the average running time to $O(n^4)$

[1] When we refer to operation counts, we assume eigenvalue computations are atomic operations with cubic complexity.

making the computation of the distance to uncontrollability for medium size systems feasible. For descriptor systems the distance to uncontrollability is discussed and a generalization of the characterization (1.10) is provided in [6].

In this work we extend the definition (1.8) for the first order system to the higher order system (1.1) as

(1.11)
$$\begin{aligned}\tau(P,B,\alpha) = \inf\{\|[\Delta K_k \ \cdots \ \Delta K_1 \ \Delta K_0 \ \Delta B]\| : \text{ the tuple} \\ (K_k+\alpha_k\Delta K_k,\ldots,K_0+\alpha_0\Delta K_0, B+\Delta B) \text{ is uncontrollable}\},\end{aligned}$$

where the vector $\alpha = [\alpha_k \ \cdots \ \alpha_1 \ \alpha_0]$ consists of nonnegative real numbers. Notice that with $k = 1$, $K_1 = I$, $K_0 = -A$, and $\alpha = [0\ 1]$ we recover the definition (1.8) for the first order system. Our motivation in introducing the scaling α is mainly to restrict the perturbations to some of the coefficient matrices, by choosing the scaling corresponding to other coefficients to be zero. It also serves the purpose of weighting the perturbations to the coefficients. For instance one may be interested in perturbations in a relative sense with respect to the norm of the coefficients in which case it is desirable to set $\alpha = [\|K_k\| \ \cdots \ \|K_1\| \ \|K_0\|]$.

The distance to uncontrollability of the higher order system defined by (1.11) and the embedded system (1.5) are related yet different quantities. The closest uncontrollable descriptor system to the embedded system would usually be obtained by perturbing the block rows of $\mathcal{A}$ other than the first one, so the resulting uncontrollable system does not correspond to an embedding of a higher order system. For instance if one of the coefficient matrices, say K_j, is considerably larger than the other coefficients as well as B in norm and $K_k^{-1}K_j$ is close to a multiple of the identity matrix, then small perturbations to the $(j+1)$th block row of $[\mathcal{A} \ \mathcal{B}]$ makes it rank deficient and the embedding uncontrollable. Typically we expect that $\tau(\mathcal{A},\mathcal{B}) < \tau(P,B,\alpha)$, since in the definition of $\tau(\mathcal{A},\mathcal{B})$ we have more degrees of freedom when choosing perturbations. Such an example where these two distances differ significantly is given in section 4.2. It is not clear how the existing algorithms to compute $\tau(\mathcal{A},\mathcal{B})$ can be modified to impose the constraints on perturbations to $\mathcal{A}$ and $\mathcal{B}$ so that perturbed systems correspond to the embeddings of higher order systems.

In the next section we provide a singular value minimization characterization for the definition (1.11). We will see that the definition (1.11) in the spectral norm and the Frobenius norm are equivalent just as in the first order case and the characterization we derive reduces to (1.10) for the first order system. The derivation of the singular value characterization uses the rank definition of the controllability (1.7) for the higher order system and all nearby systems which holds only if the leading coefficient is nonsingular and sufficiently away from the closest singular matrix. The equivalent singular value characterization is typically nonconvex. A standard optimization technique such as BFGS will converge only to a local minimum. Applying BFGS repeatedly with various starting points might occasionally fail to return a global minimum. Therefore in section 3 we describe a trisection algorithm locating the global minimum of the equivalent optimization problem. This algorithm is not a generalization of the algorithm of [3], because such an approach is too expensive. The first few steps of the new algorithm are comparatively cheap, but as we require more accuracy the algorithm becomes computationally intensive. With a complexity of $O\left(\frac{1}{\arccos(1-(\frac{tol}{k})^2)}n^3k^4\right)$ with tol denoting the accuracy required, it is devised for a few digits of precision. Section 4 is devoted to numerical examples illustrating the efficiency of the algorithm.

2. Properties of the higher order distance to uncontrollability and a singular value characterization. The set of controllable tuples is clearly a dense subset of the whole space of matrix tuples. But this does not mean that the uncontrollable tuples are isolated points. On the contrary there are uncontrollable subspaces. For instance the system (1.1) with $K_0 = 0$ and $\text{rank}(B) < n$ is uncontrollable for all $K_k, \ldots, K_1$. Therefore we shall first see that $\tau(P, B, \alpha)$ is indeed attained at some $(\Delta K_k, \ldots, \Delta K_0, \Delta B)$. Note that throughout this work we usually use $\|\cdot\|$ for either the spectral or the Frobenius norm interchangeably when the results hold for both of the norms or when the type of the norm is clear from the context. At other times we clarify the choice of norm using the notation $\|\cdot\|_2, \|\cdot\|_F$ for the spectral and the Frobenius norm, respectively.

LEMMA 2.1. [illegible] $(K_k+\alpha_k\Delta K_k, \ldots, K_0+\alpha_0\Delta K_0, B+\Delta B)$ [illegible] $\tau(P, B, \alpha) = \|[\Delta K_k \ \cdots \ \Delta K_0 \ \ \Delta B]\|$ [illegible] $\|\Delta K_j\| \leq \|B\|$ [illegible] j $\|\Delta B\| \leq \|B\|$

[illegible]. The matrix $[P(\lambda) \ \ 0]$ is rank deficient at the eigenvalues of P. Therefore $\tau(P, B, \alpha) \leq \|B\|$ meaning we can restrict the perturbations to the ones satisfying $\|\Delta K_j\| \leq \|B\|$ and $\|\Delta B\| \leq \|B\|$.

Furthermore the set of uncontrollable tuples is closed. To see this, consider any sequence $\{(K_k', \ldots, K_0', B')\}$ of uncontrollable tuples. Now for any tuple in the sequence define the associated polynomial as $P'(\lambda) = \sum_{j=0}^{k} \lambda^j K_j'$. The matrix $[P'(\lambda) \ \ B']$ is rank deficient for some λ, so all combinations of n columns of this matrix are linearly dependent. Let us denote the $l = \binom{m+n}{n}$ polynomials associated with the determinants of the combinations of n columns by $p_1(\lambda), p_2(\lambda), \ldots, p_l(\lambda)$ in any order. These polynomials must share a common root; otherwise $[P'(\lambda) \ \ B']$ would not be rank deficient for some λ. The common roots $r_1, r_2, \ldots, r_l$ are continuous functions of the tuple $\{(K_k', \ldots, K_0', B')\}$ which means at any cluster point of the sequence $r_1 = r_2 = r_3 = \cdots = r_l$. This shows that the set is closed.

Since we are minimizing the spectral or the Frobenius norm over a compact set, $\tau(P, B, \alpha)$ must be attained at some $\|[\Delta K_k \ \cdots \ \Delta K_0 \ \ \Delta B]\|$. □

The main result of this section establishes the equivalence of $\tau(P, B, \alpha)$ to the solution of the singular value minimization problem

$$\xi(P, B, \alpha) = \inf_{\lambda \in \mathbb{C}} \sigma_{\min} \left(\begin{bmatrix} P(\lambda) \\ \hline \sqrt{s_\alpha(|\lambda|)} \end{bmatrix} \ \ B \right) \tag{2.1}$$

when $\alpha_0 \neq 0$, where

$$s_\alpha(|\lambda|) = \sum_{j=0}^{k} \alpha_j^2 |\lambda|^{2j}.$$

When establishing this equivalence, we seek the perturbations ΔP and ΔB yielding a matrix function $[(P + \Delta P)(\lambda) \ B + \Delta B]$ that is rank deficient at some λ. A relevant problem is the distance to instability of a matrix polynomial which can be posed as

$$\beta(P, \alpha)$$
$$= \inf \left\{ \|[\Delta K_k \ \Delta K_{k-1} \ldots \Delta K_0]\| : (P + \Delta P)(\lambda) = 0, \ \exists \lambda \in \mathbb{C}_b, \ \Delta P = \sum_{j=0}^{k} \alpha_j \lambda^j \Delta K_j \right\}$$

where $\mathbb{C}_b$ is a closed subset of the complex plane corresponding to the unstable region and $\|\cdot\|$ is the spectral norm. A simplified version of this problem with α equal to

the vector of ones was studied in [13]. Let $\partial\mathbb{C}_b$ denote the boundary of the unstable region. It is straightforward to modify Lemma 8 in [13] to deduce the equivalence of $\beta(P,\alpha)$ with the minimization problem

$$\inf_{\lambda\in\partial\mathbb{C}_b} \sigma_{\min}\left(\left[\frac{P(\lambda)}{\sqrt{s_\alpha(|\lambda|)}}\right]\right).$$

Another similar problem is the pseudospectrum of a matrix polynomial which consists of the set of eigenvalues of nearby matrix polynomials. Let us formally define the ϵ-pseudospectrum as

$$\Lambda_\epsilon(P,\alpha) = \left\{\lambda\in\mathbb{C} : (P+\Delta P)(\lambda)=0,\ \Delta P=\sum_{j=0}^{k}\alpha_j\lambda^j\Delta K_j,\ \|[\Delta K_k\ \Delta K_{k-1}\ \cdots\ \Delta K_0]\|\le\epsilon\right\}$$

where $\|\cdot\|$ denotes the spectral norm. Here we slightly depart from the original definition suggested by Tisseur and Higham in [23] in the way the nearness to a matrix polynomial is measured. (In [23] the norm of each of the perturbations ΔK_j is constrained to be less than ϵ.) The technique in [23] leads us to the singular value characterization

$$\Lambda_\epsilon(P,\alpha) = \left\{\lambda\in\mathbb{C} : \sigma_{\min}\left(\left[\frac{P(\lambda)}{\sqrt{s_\alpha(|\lambda|)}}\right]\right)\le\epsilon\right\}.$$

The condition $\alpha_0\neq 0$, that is assumed throughout the derivations below, means that the perturbations to K_0 cannot be blocked and avoids the indeterminate case, when $s_\alpha(|\lambda|)=0$. At the end of this section we will present a more general equivalence result that holds no matter what value is assigned to α as long as all of its components are nonnegative. With this restriction on α_0, $\xi(P,B,\alpha)$ must be attained either at a finite λ or at ∞. The latter case is eliminated by the next lemma.

LEMMA 2.2. [illegible] (1.1) [illegible] $\alpha_k\xi(P,B,\alpha)$ [illegible] $\alpha_0\neq 0$ [illegible]

$$\xi(P,B,\alpha) < \lim_{\lambda\to\infty}\sigma_{\min}\left(\left[\frac{P(\lambda)}{\sqrt{s_\alpha(|\lambda|)}}\quad B\right]\right)$$

[illegible]. When $\alpha_k=0$, the result immediately follows. When $\alpha_k>0$, we have

$$\sigma_{\min}\left(\left[\frac{K_k}{\alpha_k}\quad B\right]\right) = \lim_{\lambda\to\infty}\sigma_{\min}\left(\left[\frac{P(\lambda)}{\sqrt{s_\alpha(|\lambda|)}}\quad B\right]\right).$$

Suppose $\xi(P,B,\alpha)$ is attained at ∞ and therefore there exist $u_1, v\in\mathbb{C}^n$ and $u_2\in\mathbb{C}^m$ such that

$$\begin{bmatrix}\left(\frac{K_k}{\alpha_k}\right)^*\\ B^*\end{bmatrix} v = \xi(P,B,\alpha)\begin{bmatrix}u_1\\ u_2\end{bmatrix},$$

where $[u_1^T \ u_2^T]^T$ and v have unit length. Multiplying the upper blocks by α_k, the right-hand side by v^*v and collecting all terms on the left yields

$$\begin{bmatrix} K_k^* - \alpha_k \xi(P,B,\alpha) u_1 v^* \\ B^* - \xi(P,B,\alpha) u_2 v^* \end{bmatrix} v = 0.$$

Consequently a perturbation to the leading coefficient with norm at most $\alpha_k \xi(P,B,\alpha)$ yields the singular matrix $K_k - \alpha_k \xi(P,B,\alpha) v u_1^*$, which contradicts the nonsingularity assumption. □

THEOREM 2.3. [illegible] 2.2 [illegible] (1.1) [illegible] $\tau(P,B,\alpha) = \xi(P,B,\alpha)$ [illegible] τ [illegible] (1.11) [illegible]

[illegible]. First we assume that $\tau(P,B,\alpha)$ in (1.11) is defined in the spectral norm and show that $\xi(P,B,\alpha) \leq \tau(P,B,\alpha)$. From Lemma 2.1, there exists $\Delta P(\lambda) = \sum_{j=0}^{k} \alpha_j \lambda^j \Delta K_j$ such that

$$\tau(P,B,\alpha) = \|[\Delta K_k \ \cdots \ \Delta K_0 \ \ \Delta B]\|$$

and for some $\tilde{\lambda}$ the matrix $[(P+\Delta P)(\tilde{\lambda}) \ \ B + \Delta B]$ is rank deficient, that is

$$\begin{bmatrix} ((P+\Delta P)(\tilde{\lambda}))^* \\ B^* + \Delta B^* \end{bmatrix} v = 0$$

for some unit $v \in \mathbb{C}^n$. We collect the perturbations on the right and divide the upper blocks by $\sqrt{s_\alpha(|\tilde{\lambda}|)}$ to obtain

$$\begin{bmatrix} \left(\frac{P(\tilde{\lambda})}{\sqrt{s_\alpha(|\tilde{\lambda}|)}}\right)^* \\ B^* \end{bmatrix} v = \begin{bmatrix} \left(-\frac{\Delta P(\tilde{\lambda})}{\sqrt{s_\alpha(|\tilde{\lambda}|)}}\right)^* \\ -\Delta B^* \end{bmatrix} v.$$

Therefore

$$\begin{aligned} \xi(P,B,\alpha) &\leq \sigma_{\min}\left(\begin{bmatrix} \frac{P(\tilde{\lambda})}{\sqrt{s_\alpha(|\tilde{\lambda}|)}} & B \end{bmatrix}\right) \\ &= \sigma_{\min}\left(\begin{bmatrix} \left(\frac{P(\tilde{\lambda})}{\sqrt{s_\alpha(|\tilde{\lambda}|)}}\right)^* \\ B^* \end{bmatrix}\right) \leq \left\| \begin{bmatrix} \left(\frac{P(\tilde{\lambda})}{\sqrt{s_\alpha(|\tilde{\lambda}|)}}\right)^* \\ B^* \end{bmatrix} v \right\| \\ &= \left\| \begin{bmatrix} \left(\frac{\Delta P(\tilde{\lambda})}{\sqrt{s_\alpha(|\tilde{\lambda}|)}}\right)^* \\ \Delta B^* \end{bmatrix} v \right\| \leq \left\| \begin{bmatrix} \left(\frac{\Delta P(\tilde{\lambda})}{\sqrt{s_\alpha(|\tilde{\lambda}|)}}\right)^* \\ \Delta B^* \end{bmatrix} \right\| = \left\| \begin{bmatrix} \frac{\Delta P(\tilde{\lambda})}{\sqrt{s_\alpha(|\tilde{\lambda}|)}} & \Delta B \end{bmatrix} \right\|. \end{aligned}$$

Moreover,

$$\begin{bmatrix} \frac{\Delta P(\tilde{\lambda})}{\sqrt{s_\alpha(|\tilde{\lambda}|)}} & \Delta B \end{bmatrix} = [\Delta K_k \ \cdots \ \Delta K_0 \ \ \Delta B] \begin{bmatrix} \frac{\alpha_k \tilde{\lambda}^k I}{\sqrt{s_\alpha(|\tilde{\lambda}|)}} & 0 \\ \vdots & \vdots \\ \frac{\alpha_1 \tilde{\lambda} I}{\sqrt{s_\alpha(|\tilde{\lambda}|)}} & 0 \\ \frac{\alpha_0 I}{\sqrt{s_\alpha(|\tilde{\lambda}|)}} & 0 \\ 0 & I \end{bmatrix},$$

where the spectral norm of the rightmost matrix is one. It follows from the Cauchy–Schwarz inequality that

$$\xi(P,B,\alpha) \le \left\| \begin{bmatrix} \dfrac{\Delta P(\tilde{\lambda})}{\sqrt{s_\alpha(|\tilde{\lambda}|)}} & \Delta B \end{bmatrix} \right\| \le \|[\Delta K_k \ \cdots \ \Delta K_0 \ \ \Delta B]\| = \tau(P,B,\alpha).$$

For the reverse inequality, still using the spectral norm, we have from Lemma 2.2 that for some φ,

$$\xi(P,B,\alpha) = \sigma_{\min}\left(\begin{bmatrix} \dfrac{P(\varphi)}{\sqrt{s_\alpha(|\varphi|)}} & B \end{bmatrix} \right) = \sigma_{\min}\left(\begin{bmatrix} \left(\dfrac{P(\varphi)}{\sqrt{s_\alpha(|\varphi|)}}\right)^* \\ B^* \end{bmatrix} \right)$$

or equivalently

$$\begin{bmatrix} \frac{(P(\varphi))^*}{\sqrt{s_\alpha(|\varphi|)}} \\ B^* \end{bmatrix} v = \xi(P,B,\alpha) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix},$$

where $v, u_1 \in \mathbb{C}^n$, $u_2 \in \mathbb{C}^m$, and the vectors v and $[u_1^T \ u_2^T]^T$ have unit length. We multiply the right-hand side by v^*v, the upper blocks by $\sqrt{s_\alpha(|\varphi|)}$ and collect all terms on the left to obtain

$$\begin{bmatrix} (P(\varphi))^* - \sqrt{s_\alpha(|\varphi|)}\xi(P,B,\alpha)u_1v^* \\ B^* - \xi(P,B,\alpha)u_2v^* \end{bmatrix} v = 0.$$

In other words, the matrix

$$\begin{bmatrix} P(\varphi) - \sqrt{s_\alpha(|\varphi|)}\xi(P,B,\alpha)vu_1^* & B - \xi(P,B,\alpha)vu_2^* \end{bmatrix}$$

is rank deficient. If we set $\Delta K_j = \frac{-\alpha_j \bar{\varphi}^j \xi(P,B,\alpha)vu_1^*}{\sqrt{s_\alpha(|\varphi|)}}$ and $\Delta B = -\xi(P,B,\alpha)vu_2^*$ and define $\Delta P(\lambda) = \sum_{j=0}^m \alpha_j \lambda^j \Delta K_j$, then by noting

$$\Delta P(\varphi) = \sum_{j=0}^m \alpha_j \varphi^j \Delta K_j = -\sqrt{s_\alpha(|\varphi|)}\xi(P,B,\alpha)vu_1^*$$

we see that

$$[(P+\Delta P)(\lambda) \ \ B + \Delta B]$$

is rank deficient at $\lambda = \varphi$. The norm of the perturbations satisfies

$$\begin{aligned} &\|[\Delta K_k \ \cdots \ \Delta K_0 \ \ \Delta B]\| \\ &= \xi(P,B,\alpha) \left\| \begin{bmatrix} \alpha_k \bar{\varphi}^k \dfrac{vu_1^*}{\sqrt{s_\alpha(|\varphi|)}} & \dots & \alpha_0 \dfrac{vu_1^*}{\sqrt{s_\alpha(|\varphi|)}} & vu_2^* \end{bmatrix} \right\| \le \xi(P,B,\alpha). \end{aligned}$$

Therefore $\tau(P,B,\alpha) \le \|[\Delta K_k \ \cdots \ \Delta K_0 \ \ \Delta B]\| \le \xi(P,B,\alpha)$ as desired.

For the claim about the equality when $\tau(P,B,\alpha)$ is defined in the Frobenius norm, to show $\xi(P,B,\alpha) \le \tau(P,B,\alpha)$ the proof in the first part applies noting that

$$\xi(P,B,\alpha) \le \|[\Delta K_k \ \cdots \ \Delta K_0 \ \ \Delta B]\|_2 \le \|[\Delta K_k \ \cdots \ \Delta K_0 \ \ \Delta B]\|_F = \tau(P,B,\alpha).$$

The second part to show $\tau(P,B,\alpha) \leq \xi(P,B,\alpha)$ applies without modification. ∎

The second part of Theorem 2.3 explicitly constructed the closest uncontrollable system which we state in the next corollary.

COROLLARY 2.4. [illegible] 2.2 [illegible] $\xi(P,B,\alpha)$ [illegible] λ_* [illegible] $[u_1^T\ u_2^T]^T$ [illegible] v [illegible]

$$\sigma_{\min}\left(\begin{bmatrix} \dfrac{P(\lambda_*)}{s_\alpha(|\lambda_*|)} & B \end{bmatrix}\right),$$

[illegible] $u_1, v \in \mathbb{C}^n$ [illegible] $u_2 \in \mathbb{C}^m$ [illegible] $(K_k + \alpha_k\Delta K_k, \ldots, K_0 + \alpha_0\Delta K_0, B + \Delta B)$ [illegible]

$$\Delta K_j = \frac{-\alpha_j \bar{\lambda}_*^j \xi(P,B,\alpha) v u_1^*}{\sqrt{s_\alpha(|\lambda_*|)}}, \quad j = 0, \ldots, k$$

[illegible]

$$\Delta B = -\xi(P,B,\alpha) v u_2^*.$$

Finally to remove the condition that $\alpha_0 \neq 0$, clearly $\tau(P,B,\alpha)$ depends on α_0 continuously when $\alpha_0 > 0$ and is continuous from the right when $\alpha_0 = 0$. (Consider the distance of $(K_k, K_{k-1}, \ldots, B)$ to any fixed uncontrollable tuple as a function of α_0 with all other α_j fixed. If such a distance function is bounded around a given α_0, then it is continuous from the right and the minimum of these continuous distance functions is $\tau(P,B,\alpha)$ as a function of α_0.) Therefore if $\alpha_0 = 0$, which is particularly the case when $s_\alpha(|\lambda|) = 0$, then the limiting value of $\xi(P,B,\alpha)$ from the right must approach $\tau(P,B,\alpha)$.

THEOREM 2.5. [illegible] 2.2 [illegible] α_0-[illegible] [illegible] [illegible]

$$\tau(P,B,[\alpha_k,\alpha_{k-1},\ldots,\alpha_0]) = \lim_{\alpha_0' \to \alpha_0^+} \xi(P,B,[\alpha_k,\alpha_{k-1},\ldots,\alpha_0'])$$

[illegible] τ [illegible]

Specifically when $\tau(P,B,\alpha) = \|[0\ 0 \cdots \Delta B]\| = \|\Delta B\|$, that is a closest uncontrollable system can be obtained just by perturbing B (this has to be the case when $\alpha = 0$), the result above amounts to a minimization problem over the vectors that are constrained to lie in the left eigenspace of P, $\mathcal{S}_P$, which we can see as follows. If we restrict the perturbations only to B and without loss of generality assume $\alpha = 0$, then the definition of the higher order distance to uncontrollability simplifies as

$$\begin{aligned}\tau(P,B) &= \inf\{\|\Delta B\| : v^*[P(\lambda)\ \ B + \Delta B] = 0,\ \exists v \in \mathbb{C}^n,\ \lambda \in \mathbb{C}\} \\ &= \inf\{\|\Delta B\| : v^* B = -v^* \Delta B,\ v \in \mathcal{S}_P\}.\end{aligned}$$

The last minimization problem must be attained at a ΔB such that $\|\Delta B\| = \|v^*\Delta B\|$, where $v \in \mathcal{S}_P$, because otherwise we can obtain a matrix ΔB smaller in norm by replacing all of the singular values larger than $\|v^*\Delta B\|$ with 0 that still satisfies the constraint $v^*B = -v^*\Delta B$. Therefore the last minimization problem is equivalent to

$$\tau(P,B) = \inf\{\|v^*\Delta B\| : v \in \mathcal{S}_p, v^*B = -v^*\Delta B\} = \inf_{v \in \mathcal{S}_p} \|v^*B\|.$$

Now we can verify Theorem 2.5 for this special case, as indeed

$$\lim_{\alpha_0\to 0^+} \xi(P,B,[0\ 0\ \cdots\ \alpha_0]) = \lim_{\alpha_0\to 0^+} \inf_{\lambda\in\mathbb{C}} \sigma_{\min}\left(\begin{bmatrix} P(\lambda) \\ \alpha_0 \end{bmatrix} B\right)$$
$$= \lim_{\alpha_0\to 0^+} \inf_{\lambda\in\mathbb{C}, v\in\mathbb{C}^n} \left\| v^* \begin{bmatrix} P(\lambda) \\ \alpha_0 \end{bmatrix} B \right\|.$$

Furthermore as $\alpha_0 \to 0^+$, any solution pair λ, v of the minimization problem must correspond to an eigenvalue of P and the associated left eigenvector, respectively. Therefore the minimization problem reduces to

$$\lim_{\alpha_0\to 0^+} \xi(P,B,[0\ 0\ \cdots\ \alpha_0]) = \inf_{v\in\mathcal{S}_P} \|v^*B\| = \tau(P,B).$$

3. A practical algorithm exploiting the singular value characterization. In Theorem 2.3 we established the equality

$$\tau(P,B,\alpha) = \xi(P,B,\alpha) = \inf_{r\geq 0,\theta\in[0,2\pi)} f(r,\theta)$$

when $\alpha_0 \neq 0$, where

$$f(r,\theta) = \sigma_{\min}\left(\begin{bmatrix} \dfrac{P(re^{i\theta})}{\sqrt{s_\alpha(r)}} & B \end{bmatrix}\right).$$

When $\alpha_0 = 0$, the limit of $\xi(P,B,\alpha)$ as $\alpha_0 \to 0^+$ approaches the distance to uncontrollability. Therefore, in essence the computation of the distance to uncontrollability can be achieved by minimizing $f(r,\theta)$. In this section we present a trisection algorithm to minimize the function $f(r,\theta)$ in polar coordinates. Let δ_1 and δ_2 trisect the interval $[L,U]$ containing the distance to uncontrollability (see Figure 3.1). At each iteration the algorithm updates either the upper bound to δ_1 or the lower bound to δ_2 depending on whether the δ-level set of $f(r,\theta)$

$$\{re^{i\theta} : f(r,\theta) = \delta\}$$

is intersected by any line in the set of lines passing through the origin with slopes multiples of η, where δ and η are determined by δ_1 and δ_2 as

$$\delta = \delta_1, \quad \eta = \frac{2}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta_1-\delta_2}{ckK_{\max}}\right)^2\right).$$

Above c is a positive real constant depending on the modulus of a point in the complex plane where $\xi(P,B,\alpha)$ is attained and $K_{\max}$ is a positive real constant depending on the norms of the coefficient matrices. (The constants c and $K_{\max}$ are defined precisely in the paragraph preceding Theorem 3.2.) We say the angle η subtends all of the components of the δ-level set of f, when no component has a pair of points whose angles differ by more than η. At each iteration we verify only one of the following (even though both of them may sometimes be true);

FIG. 3.1. *The trisection algorithm keeps track of an interval* $[L,U]$ *containing* $\xi(P,B,\alpha)$. *At each iteration either* L *is updated to* δ_2 *or* U *is updated to* δ_1.

- the δ-level set of f is not empty,
- the angle η subtends all of the components of the δ-level set of f.

By the definition of $\xi(P, B, \alpha)$ when the δ-level set is not empty

$$\delta = \delta_1 \geq \xi(P, B, \alpha) \tag{3.1}$$

and when η subtends all of the components of the δ-level set we will see below that

$$\xi(P, B, \alpha) > \delta_2 \tag{3.2}$$

because of the choice of η and δ. The algorithm we present is inspired by the trisection algorithm of [3] for the first order distance to uncontrollability. However, the technique we use to verify which one of (3.1) and (3.2) holds is new and has no similarity with the verification technique used in [3] to trisect an interval known to contain the first order distance to uncontrollability. A straightforward modification of the technique for the first order distance to uncontrollability would require the solution of polynomial eigenvalue problems quadratic in size and double in degree as compared to the original polynomial eigenvalue problem, which is too expensive even for systems of small size.

The trisection algorithm starts with the trivial upper bound $U = \sigma_{\min}([K_k/\alpha_k \;\; B])$ (or when $\alpha_k = 0$, $U = \sigma_{\min}(B)$) and the lower bound $L = 0$. At each iteration we either update the upper bound to δ_1 if the inequality (3.1) is verified or the lower bound to δ_2 if the inequality (3.2) is verified. First we need to be equipped with a technique that checks for a given δ and θ whether there exists an r satisfying

$$f(r, \theta) = \delta, \tag{3.3}$$

that is whether the line with slope θ passing through the origin, say $\mathcal{L}(\theta)$, intersects the δ-level set of f. Our first result in this section shows how this can be achieved by solving a polynomial eigenvalue problem of double size and of double degree. Similar results relating the δ-level set of $g(x, y) = \sigma_{\min}(A - (x + yi)I)$, where $A \in \mathbb{C}^{n\times n}$, $x, y \in \mathbb{R}$ and the imaginary eigenvalues of a matrix $G(x, \delta)$ of double size can be found in [4] and [2]. More precisely these results suggest how to find the intersection points of the δ-level set of $g(x, y)$ and a vertical line; that is the results deduce that if $\delta = g(x, y)$, then yi is an eigenvalue of $G(x, \delta)$.

THEOREM 3.1. [illegible] $\theta \in [0, 2\pi)$ [illegible] δ [illegible] $[\frac{P(re^{i\theta})}{\sqrt{s_\alpha(r)}} \;\; B]$ [illegible] δ [illegible] $Q(\lambda, \theta, \delta) = \sum_{j=0}^{2k} \lambda^j Q_j(\theta, \delta)$ [illegible] ri

$$Q_0(\theta, \delta) = \begin{bmatrix} -\delta\alpha_0^2 I & K_0^* \\ K_0 & BB^*/\delta - \delta I \end{bmatrix},$$

[illegible] l-[illegible]

$$Q_l(\theta, \delta) = \begin{bmatrix} 0 & (-1)^{(l+1)/2} i K_l^* e^{-il\theta} \\ (-1)^{(l+1)/2} i K_l e^{il\theta} & 0 \end{bmatrix} \quad 1 \leq l \leq k,$$

$$Q_l(\theta, \delta) = 0 \quad k+1 \leq l < 2k,$$

[illegible] l-[illegible]

$$Q_l(\theta, \delta) = \begin{bmatrix} (-1)^{l/2+1}\delta\alpha_{l/2}^2 I & (-1)^{l/2} K_l^* e^{-il\theta} \\ (-1)^{l/2} K_l e^{il\theta} & 0 \end{bmatrix} \quad 1 \leq l \leq k,$$

$$Q_l(\theta, \delta) = \begin{bmatrix} (-1)^{l/2+1}\delta\alpha_{l/2}^2 I & 0 \\ 0 & 0 \end{bmatrix} \quad k+1 \leq l \leq 2k.$$

. The matrix $[\frac{P(re^{i\theta})}{\sqrt{s_\alpha(r)}} \;\; B]$ has δ as a singular value if and only if both of the equations

$$\begin{bmatrix} \frac{P(re^{i\theta})}{\sqrt{s_\alpha(r)}} & B \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \delta u,$$
$$\begin{bmatrix} \left(\frac{P(re^{i\theta})}{\sqrt{s_\alpha(r)}}\right)^* \\ B^* \end{bmatrix} u = \delta \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

are satisfied. From the bottom block of the second equation we have $v_2 = B^*u/\delta$. By eliminating v_2 from the other equation, we obtain

$$\begin{aligned} &\begin{bmatrix} -\delta I & \left(\frac{P(re^{i\theta})}{\sqrt{s_\alpha(r)}}\right)^* \\ \frac{P(re^{i\theta})}{\sqrt{s_\alpha(r)}} & BB^*/\delta - \delta I \end{bmatrix} \begin{bmatrix} v_1 \\ u \end{bmatrix} \\ &= \begin{bmatrix} -\delta s_\alpha(r) I & \left(P(re^{i\theta})\right)^* \\ P(re^{i\theta}) & BB^*/\delta - \delta I \end{bmatrix} \begin{bmatrix} v_1/\sqrt{s_\alpha(r)} \\ u \end{bmatrix} \\ &= \sum_{j=0}^{2k} (ri)^j Q_j(\theta, \delta) \begin{bmatrix} v_1/\sqrt{s_\alpha(r)} \\ u \end{bmatrix} = 0. \end{aligned}$$

Therefore ri is an eigenvalue of $Q(\lambda, \theta, \delta)$. □

Suppose $\delta \le \lim_{\lambda\to\infty} \sigma_{\min}([\frac{P(\lambda)}{\sqrt{s_\alpha(|\lambda|)}} \;\; B])$. To establish the existence of an r satisfying (3.3), it is sufficient that the polynomial $Q(\lambda, \theta, \delta)$ has an imaginary eigenvalue. When $Q(\lambda, \theta, \delta)$ has an imaginary eigenvalue $r'i$, $f(r', \theta) \le \delta$. Since $\delta \le f(r, \theta)$ in the limit as $r \to \infty$, by the continuity of f with respect to r we deduce $f(\hat{r}, \theta) = \delta$ for some $\hat{r} \ge r'$.

For our trisection algorithm it suffices to check whether any of the lines $\mathcal{L}(0), \mathcal{L}(\eta), \mathcal{L}(2\eta), \ldots, \mathcal{L}(\lfloor \frac{\pi}{\eta} \rfloor \eta)$ intersect the δ-level set of f as illustrated in Figure 3.2. When there is an intersection point the δ-level set is not empty; otherwise the angle η subtends all of the components. The only part of the algorithm that is not clarified so far is how we conclude a lower bound on $\xi(P, B, \alpha)$ when η subtends all of the components, in particular the relation between δ_2 in (3.2) and the pair δ and η. For the next theorem addressing these issues let (r_*, θ_*) be a point where $\xi(P, B, \alpha)$ is attained. We assume the existence of a constant c known , , satisfying

$$c \ge \max_{0\le j\le k} \frac{r_*^j}{\sqrt{s_\alpha(r_*)}} = \max\left(\frac{1}{\sqrt{s_\alpha(r_*)}}, \frac{r_*^k}{\sqrt{s_\alpha(r_*)}}\right). \tag{3.4}$$

Finding a constant c may be tedious in some special cases. However, when both α_k and α_0 are nonzero we can set $c = \frac{1}{\min(\alpha_0, \alpha_k)}$. We furthermore use the notation $K_{\max} = \max_{1\le j\le k} \|K_j\|$. The algorithms in [14, 15, 3] for the first order distance to uncontrollability benefit from an analogous result in [14] which can be stated as, given a $\delta \ge \tau(A, B)$ for all $\eta \in [0, 2(\delta - \tau(A, B))]$ there exists a pair of real numbers x, y satisfying $\sigma_{\min}([A - (x + yi)I \;\; B]) = \sigma_{\min}([A - (x + \eta + yi)I \;\; B]) = \delta$. Throughout the rest of this section we omit the parameters of $\xi(P, B, \alpha)$ assuming P, B, and α are fixed.

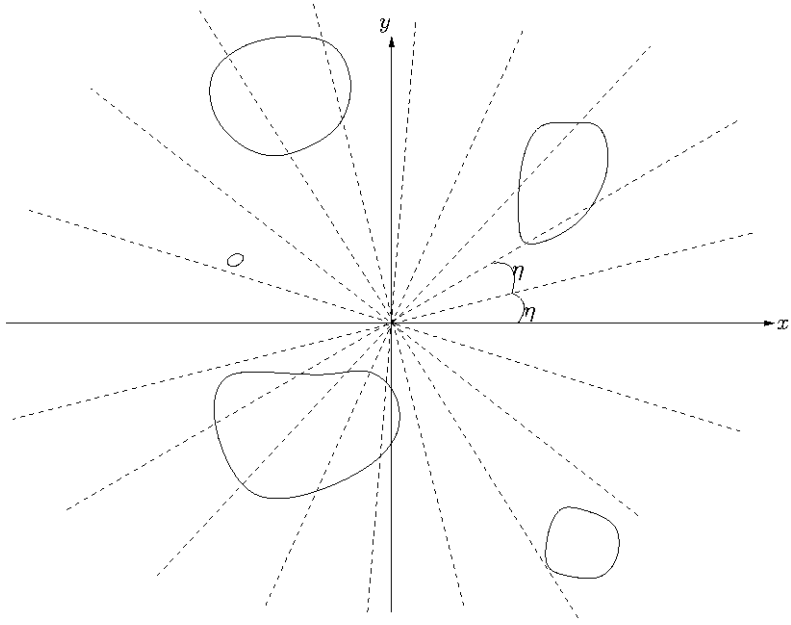


FIG. 3.2. *To verify which one of* (3.1) *and* (3.2) *hold we check the intersection points of the* δ*-level set of* f *and the set of lines with slopes multiples of* η *ranging from 0 to* π*. The closed curves are the* δ*-level sets.*

THEOREM 3.2.

$$\lim_{\lambda\to\infty} \sigma_{\min}\left(\left[\frac{P(\lambda)}{\sqrt{s_\alpha(|\lambda|)}} \;\; B\right]\right) \geq \delta > \xi.$$

$\eta \in [0, \frac{1}{k}\arccos(1-\frac{1}{2}(\frac{\delta-\xi}{ckK_{\max}})^2)]$ r_1 r_2 (η)

$$\sigma_{\min}\left(\left[\frac{P(r_1 e^{i(\theta_*+\eta)})}{\sqrt{s_\alpha(r_1)}} \;\; B\right]\right) = \delta \quad \text{and} \quad \sigma_{\min}\left(\left[\frac{P(r_2 e^{i(\theta_*-\eta)})}{\sqrt{s_\alpha(r_2)}} \;\; B\right]\right) = \delta.$$

. We prove the first equality. The proof of the second equality is similar. Assume

$$\sigma_{\min}\left(\left[\frac{P(re^{i(\theta_*+\eta)})}{\sqrt{s_\alpha(r)}} \;\; B\right]\right) > \delta \tag{3.5}$$

holds for all r for an η in the interval specified. Since the singular values of a matrix X are the eigenvalues of the symmetric matrix

$$\begin{bmatrix} 0 & X \\ X^* & 0 \end{bmatrix},$$

they are globally Lipschitz with constant 1 (see Weyl's Theorem [19, Theorem (4.3.1)]) meaning

$$\delta - \xi < \sigma_{\min}\left(\begin{bmatrix} \frac{P(r_* e^{i(\theta_*+\eta)})}{\sqrt{s_\alpha(r_*)}} & B \end{bmatrix}\right) - \sigma_{\min}\left(\begin{bmatrix} \frac{P(r_* e^{i\theta_*})}{\sqrt{s_\alpha(r_*)}} & B \end{bmatrix}\right)$$
$$\leq \left\| \begin{bmatrix} \frac{P(r_* e^{i(\theta_*+\eta)})}{\sqrt{s_\alpha(r_*)}} & B \end{bmatrix} - \begin{bmatrix} \frac{P(r_* e^{i\theta_*})}{\sqrt{s_\alpha(r_*)}} & B \end{bmatrix} \right\| = \left\| \frac{\sum_{j=1}^{k} r_*^j e^{ij\theta_*} K_j (e^{ij\eta} - 1)}{\sqrt{s_\alpha(r_*)}} \right\|.$$

Notice that $\eta \leq \pi/k$ implying $\cos k\eta \leq \cos j\eta$ for $j = 0, \ldots, k$. Therefore

$$kcK_{\max}\sqrt{2 - 2\cos k\eta} \geq \sum_{j=1}^{k} c\|K_j\sqrt{2 - 2\cos j\eta}\| \geq \left\| \frac{\sum_{j=1}^{k} r_*^j e^{ij\theta_*} K_j (e^{ij\eta} - 1)}{\sqrt{s_\alpha(r_*)}} \right\| > \delta - \xi$$

or

$$1 - \frac{1}{2}\left(\frac{\delta - \xi}{kcK_{\max}}\right)^2 > \cos k\eta.$$

Since the $\cos$ function is strictly decreasing in the interval $[0, \pi]$, we obtain the contradiction that

$$\eta > \frac{1}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta - \xi}{kcK_{\max}}\right)^2\right).$$

Thus, (3.5) cannot hold, so there exists r_1' satisfying

$$\sigma_{\min}\left(\begin{bmatrix} \frac{P(r_1' e^{i(\theta_*+\eta)})}{\sqrt{s_\alpha(r_1')}} & B \end{bmatrix}\right) \leq \delta.$$

The first equality must therefore hold for some $r_1 \geq r_1'$ because of the continuity of $f(r, \theta_* + \eta)$ with respect to r and the fact that $\lim_{r\to\infty} f(r, \theta_* + \eta) \geq \delta$. □

As we have already indicated in (3.1), we first set $\delta = \delta_1$. The assignment

$$\eta = \frac{2}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta_1 - \delta_2}{ckK_{\max}}\right)^2\right) \tag{3.6}$$

leads us to the lower bound (3.2) in the case that none of the lines $\mathcal{L}(0), \mathcal{L}(\eta), \mathcal{L}(2\eta), \ldots, \mathcal{L}(\lfloor \frac{\pi}{\eta} \rfloor \eta)$ intersect the δ-level set of f, which we can see as follows. According to Theorem 3.2 for all θ in the interval

$$\left[\theta_* - \frac{1}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta - \xi}{ckK_{\max}}\right)^2\right), \theta_* + \frac{1}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta - \xi}{ckK_{\max}}\right)^2\right)\right], \tag{3.7}$$

the line $\mathcal{L}(\theta)$ intersects the δ-level set of f. When none of the lines $\mathcal{L}(0), \mathcal{L}(\eta), \mathcal{L}(2\eta), \ldots, \mathcal{L}(\lfloor \frac{\pi}{\eta} \rfloor \eta)$ intersects the δ-level set of f, it follows that η must be greater than the length of the interval in (3.7), that is

$$\eta = \frac{2}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta_1 - \delta_2}{ckK_{\max}}\right)^2\right) > \frac{2}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta - \xi}{ckK_{\max}}\right)^2\right).$$

From this inequality it is straightforward to deduce the lower bound (3.2). Algorithm 1 summarizes the approach described.

As the accuracy and efficiency of the algorithm depend on the extraction of the imaginary eigenvalues of the matrix polynomial $Q(\lambda, \theta, \delta)$, it is worth pointing out how these eigenvalues can be computed numerically in a reliable fashion. The matrix polynomial $Q(\lambda, \theta, \delta)$ has a special structure; its even coefficients are Hermitian, while its odd coefficients are skew-Hermitian. The eigenvalues of polynomials with this structure are either imaginary or in pairs $(\lambda, -\bar{\lambda})$ [20]. The standard way to solve a polynomial eigenvalue problem of size $2n$ and degree $2k$ is to reduce it to an equivalent generalized eigenvalue problem $\mathcal{H} - \lambda\mathcal{N}$ of size $4nk$ by a transformation called linearization. The most widely used linearization is the companion form [21]. In [21] vector spaces of linearizations that are generalizations of the companion form are introduced. There are two issues one needs to consider when selecting a linearization. First the structure must be preserved, that is the matrices $\mathcal{H}, \mathcal{N}$ in the transformation above must be Hermitian and skew-Hermitian, respectively. Sec-

Algorithm 1 Trisection algorithm for the higher order distance to uncontrollability

Call:	$[L, U] \leftarrow$ `HODU`(P,B,α,tol,c).
Input:	$P \in \mathbb{C}^{k\times n\times n}$ (the matrix polynomial), $B \in \mathbb{C}^{n\times m}$, $\alpha \in \mathbb{R}^k$ (nonnegative scaling factors, not all zero), tol (desired tolerance), c (a positive real number satisfying (3.4)).
Output:	L, U with $L < U$, $U - L \leq tol$. The interval $[L, U]$ contains the higher order distance to uncontrollability.

Initially set

$$U \leftarrow \sigma_{\min}\left(\begin{bmatrix} \frac{K_k}{\alpha_k} & B \end{bmatrix}\right) \quad \text{if } \alpha_k > 0,$$
$$U \leftarrow \sigma_{\min}(B) \quad \text{if } \alpha_k = 0,$$

and $L \leftarrow 0$.
while $U - L > tol$ **do**
  % Trisection step
  Set $\delta_1 \leftarrow L + 2(U - L)/3$ and $\delta_2 \leftarrow L + (U - L)/3$.
  Set $\delta \leftarrow \delta_1$ and η as defined in (3.6)
  Set $Intersection \leftarrow FALSE$.
  for $\theta = 0$ to π in increments of η **do**
    Compute the eigenvalues of $Q(\lambda, \theta, \delta)$.
    if $Q(\lambda, \theta, \delta)$ has an imaginary eigenvalue **then**
      % An intersection point is detected
      Update the upper bound, $U \leftarrow \delta_1$.
      $Intersection \leftarrow TRUE$.
      Break. (Leave the for loop.)
    end if
  end for
  if $\neg Intersection$ **then**
    % No intersection point is detected
    Update the lower bound, $L \leftarrow \delta_2$.
  end if
end while
Return $[L, U]$.

ond the eigenvalues of the pencil $\mathcal{H} - \lambda\mathcal{N}$ have different condition numbers than the eigenvalues of the matrix polynomial $Q(\lambda, \theta, \delta)$. Ideally we must use a linearization preserving the structure that does not degrade the conditioning of the eigenvalues of the original problem. The linearizations in the vector spaces specified in [21] that preserve the even-odd structure of $Q(\lambda, \theta, \delta)$ are identified in [20]. Furthermore in [17] it was shown that in these vector spaces there are linearizations preserving the conditioning of the eigenvalues of $Q(\lambda, \theta, \delta)$. How best to find such a linearization preserving the structure and the conditioning combined with an even-odd generalized eigenvalue solver is still under investigation. When such an implementation is used, simple imaginary eigenvalues remain on the imaginary axis even in the presence of rounding errors. Therefore tolerances are not needed.

At each iteration the algorithm requires the solution of the eigenvalue problems $Q(\lambda, 0, \delta), Q(\lambda, \eta, \delta), \ldots, Q(\lambda, \lfloor \frac{\pi}{\eta} \rfloor \eta, \delta)$, each typically at a cost of $O(n^3k^3)$. The overall complexity of an iteration is

$$O\left(\frac{n^3k^4}{\arccos\left(1 - \frac{1}{2}\left(\frac{\delta_1 - \delta_2}{ckK_{\max}}\right)^2\right)}\right). \tag{3.8}$$

It is apparent that the initial iterations for which $\delta_1 - \delta_2$ is relatively large are cheaper, while the last iteration for which $\delta_1 - \delta_2 \approx tol/2$ is the most expensive.

4. Numerical results. All of the numerical experiments in this section are performed with MATLAB 6.5 running on a PC with 1000 MHz Intel processor and 256MB RAM.

4.1. Computing the distance to uncontrollability for first order systems. Even though it is much slower than the methods in [14, 3, 15], the trisection algorithm suggested can be applied to estimate the first order distance to uncontrollability with $k = 1$, $K_1 = I$, and $\alpha = [0\ 1]$ so that perturbations to $K_1 = I$ are not allowed. It is well known that in this case the distance to uncontrollability is attained at a point λ_* with $|\lambda_*| = c \le 2(\|K_0\| + \|B\|)$. We choose K_0 as the Toeplitz matrix

$$\begin{bmatrix} 1 & 3 & 0 & 0 \\ -2 & 1 & 3 & 0 \\ 0 & -2 & 1 & 3 \\ 0 & 0 & -2 & 1 \end{bmatrix}$$

and $B = [2\ 2\ 2\ 2]^T$. When we require an interval of length 10^{-2} or less, Algorithm 1 returns $[0.473, 0.481]$ in 12 iterations which contains the distance to uncontrollability 0.477. Table 4.1 lists the cumulative running time after each iteration in seconds. Overall we observe that reaching one digit accuracy is considerably cheaper than two digit accuracy. When we allow the perturbations to the leading coefficient by setting $\alpha = [1\ 1]$, there is a closer uncontrollable system at a distance of $\tau(P, B, \alpha) \le 0.145$ which is the upper bound returned by Algorithm 1.

4.2. A quadratic brake model. In [12] the vibrations of a drum brake system are modeled by the quadratic equation

$$Mx^{(2)}(t) + K(\mu)x(t) = f(t) \tag{4.1}$$

TABLE 4.1
Total running time of the trisection algorithm after each iteration on a Toeplitz matrix and a vector pair.

Iteration	**Total running time**	**Interval** $[L, U]$
1	0.400	[0.000,0.667]
2	1.680	[0.222,0.667]
3	2.510	[0.222,0.519]
4	5.369	[0.321,0.519]
5	9.670	[0.387,0.519]
6	16.110	[0.431,0.519]
7	20.140	[0.431,0.489]
8	34.580	[0.450,0.489]
9	56.770	[0.463,0.489]
10	70.470	[0.463,0.481]
11	118.40	[0.469,0.481]
12	190.93	[0.473,0.481]

TABLE 4.2
The intervals computed by the trisection algorithm for the brake system for various μ values in an absolute sense in the second column and in a relative sense in the third column.

μ	**Interval** $[L, U]$ **(Absolute)**	**Interval** $[L, U]$ **(Relative)**
0.05	[0.051,0.059]	[0.038,0.046]
0.10	[0.097,0.105]	[0.071,0.079]
0.15	[0.140,0.148]	[0.104,0.112]
0.20	[0.184,0.191]	[0.137,0.145]
0.50	[0.418,0.426]	[0.325,0.333]
1	[0.676,0.684]	[0.574,0.581]
10	[0.990,0.997]	[0.984,0.991]
100	[0.993,1.000]	[0.987,0.994]
1000	[0.993,1.000]	[0.987,0.994]

with the mass and stiffness matrices

$$M = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix}, \quad K(\mu) = g \begin{bmatrix} (\sin\gamma + \mu\cos\gamma)\sin\gamma & -\mu - (\sin\gamma + \mu\cos\gamma)\cos\gamma \\ (\mu\sin\gamma - \cos\gamma)\sin\gamma & 1 + (-\mu\sin\gamma + \cos\gamma)\cos\gamma \end{bmatrix}.$$

Suppose the force on the brake system has just the vertical component determined by the input

$$f(t) = \begin{bmatrix} f_x(t) \\ f_y(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t).$$

For the parameters $m = 5$, $g = 1$ and $\gamma = \frac{\pi}{100}$, we consider two cases. First by setting $\alpha = [1\ 0\ 1]$, we impose equal importance on the perturbations to the mass and stiffness matrices. Notice that for small μ and γ, the system is close to being uncontrollable. In the second column in Table 4.2 the intervals of length 10^{-2} or less containing the distance to uncontrollability returned by Algorithm 1 are provided for various values of μ. The algorithm iterates 16 times to reach two digit accuracy. Second we assign scaling to the perturbations proportional to the norms of the mass and stiffness matrices, that is $\alpha = [\|M\|\ 0\ \|K\|]$. The intervals returned by Algorithm 1 for this second case are given in the rightmost column in Table 4.2. As expected the distance to uncontrollability again increases with respect to μ. The system (4.1) is closer to being uncontrollable in a relative sense than in an absolute sense.

If we allow perturbations to all coefficients with equal scaling (e.g., $\alpha = [1\ 1\ 1]$), then usually the first order distance uncontrollability of the embedded system (1.5) is

TABLE 4.3
Running time of the trisection algorithm in seconds with respect to the size and order of the systems with normally distributed coefficient matrices.

Size / order	First order	Quadratic	Cubic
5	10 (10)	192 (12)	1237 (13)
10	83 (12)	1392 (11)	12485 (12)
15	271 (13)	6390 (14)	37324 (12)

considerably smaller than the actual value $\tau(P, B, \alpha)$, since the perturbations are not constrained so that the structure of the embedding can be preserved. For instance, for the drum brake system with $\alpha = [1\ 1\ 1]$ and $\mu = 0.1$, $\tau(P, B, \alpha) \in [0.097, 0.105]$ (up to two digit accuracy it does not make any difference whether we allow perturbations to the zero coefficient K_1 or not) whereas the standard unstructured distance to uncontrollability of the embedding lies in the interval $[0.012, 0.020]$.

4.3. Running time with respect to the size and order of the system. We run the trisection algorithm on systems with random coefficients of various size and order. To be precise the entries of all of the coefficient matrices are chosen from a normal distribution with zero mean and variance one independently. Table 4.3 illustrates how the running time in seconds varies with respect to the size and order of the system. In all of the examples intervals of length at most 10^{-2} containing the absolute distance to uncontrollability (α is the vector of ones) are returned. The numbers in parentheses correspond to the number of trisection iterations needed. The variation in the running time with respect to the size and order is consistent with the complexity suggested by (3.8).

Acknowledgments. A MATLAB implemention of the trisection algorithm is available on the author's web page.[2] Most of this work was completed during the author's Ph.D. study at New York University and some part was completed during the author's visit to the numerical analysis and modeling group at the Technical University of Berlin. The author is grateful to Michael Overton and Daniel Kressner for reading a preliminary version of this paper, Volker Mehrmann for pointing out the importance of the even-odd matrix polynomials and insightful discussions regarding preserving the even-odd structure, and two anonymous referees.

REFERENCES

[1] D. BOLEY, *Estimating the sensitivity of the algebraic structure of pencils with simple eigenvalue estimates*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 632–643.
[2] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Optimization and pseudospectra, with applications to robust stability*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 80–104.
[3] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Pseudospectral components and the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 350–361.
[4] R. BYERS, *A bisection method for measuring the distance of a stable matrix to the unstable matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 875–881.
[5] R. BYERS, *Detecting nearly uncontrollable pairs*, in Proceedings of the International Symposium MTNS-89, vol. III, Amsterdam, 1989, Progr. Systems Control Theory 5, M. A. Kaashoek, J. H. van Schuppen, and A. C. M. Ran, eds., Birkhäuser Boston, Boston, 1990, pp. 447–457.
[6] R. BYERS, *The descriptor controllability radius*, in Proceedings of the International Symposium MTNS-93, Regensburg, Germany, vol. II, U. Helmke, R. Mennicken, and H. Saurer, eds., Akademie Verlag, Berlin, 1994, pp. 85–88.

[2] http://www.cs.nyu.edu/mengi/robust_stability/dist_uncont_higher.html

[7] K-W. Chu, *Controllability of descriptor systems*, Internat. J. Control, 46 (1987), pp. 1761–1770.
[8] K-W. Chu, *A controllability condensed form and a state feedback pole assignment algorithm or descriptor systems*, IEEE Trans. Automat. Control, 33 (1988), pp. 366–370.
[9] G. E. Dullerud and F. Paganini, *A Course in Robust Control Theory: A Convex Approach*, Springer-Verlag, New York, 2000.
[10] R. Eising, *Between controllable and uncontrollable*, Systems Control Lett., 4 (1984), pp. 263–264.
[11] M. Gao and M. Neumann, *A global minimum search algorithm for estimating the distance to uncontrollability*, Linear Algebra Appl., 188/189 (1993), pp. 305–350.
[12] L. Gaul and N. Wagner, *Eigenpath Dynamics of Nonconservative Mechanical Systems Such as Disc Brakes*, in IMAC XXII, Dearborn, MI, 2004.
[13] Y. Genin, R. Stefan, and P. Van Dooren, *Real and complex stability radii of polynomial matrices*, Linear Algebra Appl., 351/352 (2002), pp. 381–410.
[14] M. Gu, *New methods for estimating the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 989–1003.
[15] M. Gu, E. Mengi, M. L. Overton, J. Xia, and J. Zhu, *Fast methods for estimating the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 477–502.
[16] C. He, *Estimating the distance to uncontrollability: A fast method and a slow one*, Systems Control Lett., 26 (1995), pp. 275–281.
[17] N. J. Higham, D. S. Mackey, and F. Tisseur, *The conditioning of linearizations of matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1005–1028.
[18] N. J. Higham and F. Tisseur, *More on pseudospectra for polynomial eigenvalue problems and applications in control theory*, Linear Algebra Appl., 351/352 (2002), pp. 435–453.
[19] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
[20] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Structured polynomial eigenvalue problems: Good vibrations from good linearizations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1029–1051.
[21] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 971–1004.
[22] C. C. Paige, *Properties of numerical algorithms relating to computing controllability*, IEEE Trans. Automat. Control, 26 (1981), pp. 130–138.
[23] F. Tisseur and N. J. Higham, *Structured pseudospectra for polynomial eigenvalue problems with applications*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 187–208.
[24] M. Wicks and R. A. DeCarlo, *Computing the distance to an uncontrollable system*, IEEE Trans. Automat. Control, 36 (1991), pp. 39–49.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 173–196

AN EQUILIBRIUM PROBLEM FOR THE LIMITING EIGENVALUE DISTRIBUTION OF BANDED TOEPLITZ MATRICES*

MAURICE DUITS† AND ARNO B. J. KUIJLAARS‡

Abstract. We study the limiting eigenvalue distribution of $n \times n$ banded Toeplitz matrices as $n \to \infty$. From classical results of Schmidt, Spitzer, and Hirschman it is known that the eigenvalues accumulate on a special curve in the complex plane and the normalized eigenvalue counting measure converges weakly to a measure on this curve as $n \to \infty$. In this paper, we characterize the limiting measure in terms of an equilibrium problem. The limiting measure is one component of the unique vector of measures that minimizes an energy functional defined on admissible vectors of measures. In addition, we show that each of the other components is the limiting measure of the normalized counting measure on certain generalized eigenvalues.

Key words. banded Toeplitz matrix, equilibrium problem, potential theory, limiting eigenvalue distribution, generalized eigenvalues

AMS subject classifications. 15A18, 30E20, 31A99, 47B06

DOI. 10.1137/070687141

1. Introduction.

For an integrable function $a : \{z \in \mathbb{C} \mid |z| = 1\} \to \mathbb{C}$ defined on the unit circle in the complex plane, the $n \times n$ Toeplitz matrix $T_n(a)$ with symbol a is defined by

$$\big(T_n(a)\big)_{jk} = a_{j-k}, \qquad j,k = 1,\dots,n, \tag{1.1}$$

where a_k is the kth Fourier coefficient of a,

$$a_k = \frac{1}{2\pi}\int_0^{2\pi} a(\mathrm{e}^{\mathrm{i}\theta})\mathrm{e}^{-\mathrm{i}k\theta}\,\mathrm{d}\theta. \tag{1.2}$$

In this paper we study banded Toeplitz matrices for which the symbol has only a finite number of nonzero Fourier coefficients. We assume that there exist $p, q \geq 1$ such that

$$a(z) = \sum_{k=-q}^{p} a_k z^k, \qquad a_p \neq 0, \quad a_{-q} \neq 0. \tag{1.3}$$

Thus $T_n(a)$ has at most $p + q + 1$ nonzero diagonals. As in [1, page 263], we also assume, without loss of generality, that

$$\gcd\{k \in \mathbb{Z} \mid a_k \neq 0\} = 1. \tag{1.4}$$

*Received by the editors April 2, 2007; accepted for publication by H. J. Woerdeman July 26, 2007; published electronically February 22, 2008. This research was supported by the European Science Foundation Program MISGAM.
http://www.siam.org/journals/simax/30-1/68714.html

†Department of Mathematics, Katholieke Universiteit Leuven, Celestijnenlaan 200B, 3001 Leuven, Belgium, and Fund for Scientific Research-Flanders (maurice.duits@wis.kuleuven.be).

‡Department of Mathematics, Katholieke Universiteit Leuven, Celestijnenlaan 200B, 3001 Leuven, Belgium (arno.kuijlaars@wis.kuleuven.be). This author was supported by FWO-Flanders project G.0455.04, by K.U. Leuven research grant OT/04/21, by Belgian Interuniversity Attraction Pole NOSY P06/02, and by a grant from the Ministry of Education and Science of Spain, project code MTM2005-08648-C02-01.

We are interested in the limiting behavior of the spectrum of $T_n(a)$ as $n \to \infty$. We use $\operatorname{sp} T_n(a)$ to denote the spectrum of $T_n(a)$:

$$\operatorname{sp} T_n(a) = \{\lambda \in \mathbb{C} \mid \det(T_n(a) - \lambda I) = 0\}.$$

Spectral properties of banded Toeplitz matrices are the topic of the recent book [1] by Böttcher and Grudsky. We will refer the reader to this book frequently, in particular to Chapter 11 where the limiting behavior of the spectrum is discussed.

The limiting behavior of $\operatorname{sp} T_n(a)$ was characterized by Schmidt and Spitzer [10]. They considered the set

$$\liminf_{n\to\infty} \operatorname{sp} T_n(a) \tag{1.5}$$

consisting of all $\lambda \in \mathbb{C}$ such that there exists a sequence $\{\lambda_n\}_{n\in\mathbb{N}}$, with $\lambda_n \in \operatorname{sp} T_n(a)$, converging to λ, and the set

$$\limsup_{n\to\infty} \operatorname{sp} T_n(a) \tag{1.6}$$

consisting of all λ such that there exists a sequence $\{\lambda_n\}_{n\in\mathbb{N}}$, with $\lambda_n \in \operatorname{sp} T_n(a)$, that has a subsequence converging to λ. Schmidt and Spitzer showed that these two sets are equal and can be characterized in terms of the algebraic equation

$$a(z) - \lambda = \sum_{k=-q}^{p} a_k z^k - \lambda = 0. \tag{1.7}$$

For every $\lambda \in \mathbb{C}$ there are $p + q$ solutions for (1.7), which we denote by $z_j(\lambda)$ for $j = 1, \ldots, p + q$. We order these solutions by absolute value, so that

$$0 < |z_1(\lambda)| \le |z_2(\lambda)| \le \cdots \le |z_{p+q}(\lambda)|. \tag{1.8}$$

When all inequalities in (1.8) are strict then the values $z_k(\lambda)$ are unambiguously defined. If equalities occur, then we choose an arbitrary numbering so that (1.8) holds. The result by Schmidt and Spitzer [10], [1, Theorem 11.17] is that

$$\liminf_{n\to\infty} \operatorname{sp} T_n(a) = \limsup_{n\to\infty} \operatorname{sp} T_n(a) = \Gamma_0, \tag{1.9}$$

where

$$\Gamma_0 := \{\lambda \in \mathbb{C} \mid |z_q(\lambda)| = |z_{q+1}(\lambda)|\}. \tag{1.10}$$

This result gives a description of the asymptotic location of the eigenvalues. The eigenvalues accumulate on the set Γ_0, which is known to be a disjoint union of a finite number of (open) analytic arcs and a finite number of exceptional points [1, Theorem 11.9]. It is also known that Γ_0 is connected (see [13], [1, Theorem 11.19]) and that $\mathbb{C} \setminus \Gamma_0$ need not be connected (see [1, Theorem 11.20], [2, Proposition 5.2]). See [1] for many beautiful illustrations of eigenvalues of banded Toeplitz matrices.

The limiting eigenvalue distribution was determined by Hirschman [5], [1, Theorem 11.16]. He showed that there exists a Borel probability measure μ_0 on Γ_0 such that the normalized eigenvalue counting measure of $T_n(a)$ converges weakly to μ_0 as $n \to \infty$. That is,

$$\frac{1}{n} \sum_{\lambda \in \operatorname{sp} T_n(a)} \delta_\lambda \to \mu_0, \tag{1.11}$$

where in the sum each eigenvalue is counted according to its multiplicity. The measure μ_0 is absolutely continuous with respect to the arclength measure on Γ_0 and has an analytic density on each open analytic arc in Γ_0, which can be explicitly represented in terms of the solutions of the algebraic equation (1.7) as follows. Equip every open analytic arc in Γ_0 with an orientation. The orientation induces $\pm$ sides on each arc, where the $+$ side is on the left when traversing the arc according to its orientation, and the $-$ side is on the right. The limiting measure μ_0 is then given by

$$\mathrm{d}\mu_0(\lambda) = \frac{1}{2\pi \mathrm{i}} \sum_{j=1}^{q} \left(\frac{z'_{j+}(\lambda)}{z_{j+}(\lambda)} - \frac{z'_{j-}(\lambda)}{z_{j-}(\lambda)} \right) \mathrm{d}\lambda, \tag{1.12}$$

where $\mathrm{d}\lambda$ is the complex line element on Γ_0 (taken according to the orientation), and where $z_{j\pm}(\lambda)$, $\lambda \in \Gamma_0$, is the limiting value of $z_j(\lambda')$ as $\lambda' \to \lambda$ from the $\pm$ side of the arc. These limiting values exist for every $\lambda \in \Gamma_0$, with the possible exception of the finite number of exceptional points.

Note that the right-hand side of (1.12) is a priori a complex measure, and it is not immediately clear that it is in fact a probability measure. In the original paper [5] and in the book [1, Theorem 11.16], the authors give a different expression for the limiting density, from which it is clear that the measure is nonnegative. We prefer to work with the complex expression (1.12), since it allows for a direct generalization which we will need in this paper.

Note also that if we reverse the orientation on an arc in Γ_0, then the $\pm$ sides are reversed. Since the complex line element $\mathrm{d}\lambda$ changes sign as well, the expression (1.12) does not depend on the choice of orientation.

The following is a very simple example, which, however, serves as a motivation for the results in the paper.

EXAMPLE 1.1. Consider the symbol $a(z) = z + 1/z$. In this case we find that $\Gamma_0 = [-2, 2]$ and μ_0 is absolutely continuous with respect to the Lebesgue measure and has density

$$\frac{\mathrm{d}\mu_0(\lambda)}{\mathrm{d}\lambda} = \frac{1}{\pi\sqrt{4-\lambda^2}}, \qquad \lambda \in (-2, 2). \tag{1.13}$$

This measure is well known in potential theory and is called the arcsine measure or the equilibrium measure of Γ_0; see, e.g., [9]. It has the property that it minimizes the energy functional I, defined by

$$I(\mu) = \iint \log \frac{1}{|x-y|} \, \mathrm{d}\mu(x) \, \mathrm{d}\mu(y), \tag{1.14}$$

among all Borel probability measures μ on $[-2, 2]$. The measure μ_0 is also characterized by the equilibrium condition

$$\int \log|x-\lambda| \, \mathrm{d}\mu_0(\lambda) = 0, \qquad x \in [-2, 2], \tag{1.15}$$

which is the Euler–Lagrange variational condition for the minimization problem.

The fact that μ_0 is the equilibrium measure of Γ_0 is special for symbols a with $p = q = 1$. In that case one may think of the eigenvalues of $T_n(a)$ as charged particles on Γ_0, each eigenvalue having a total charge $1/n$, that repel each other with logarithmic interaction. The particles seek to minimize the energy functional (1.14).

As $n \to \infty$, they distribute themselves according to μ_0, and μ_0 is the minimizer of (1.14) among all probability measures supported on Γ_0.

The aim of this paper is to characterize μ_0 for general symbols a of the form (1.3) also in terms of an equilibrium problem from potential theory. The corresponding equilibrium problem is more complicated since it involves not only the measure μ_0, but a sequence of $p+q-1$ measures

$$\mu_{-q+1}, \mu_{-q+2}, \ldots, \mu_{-1}, \mu_0, \mu_1, \ldots, \mu_{p-2}, \mu_{p-1}$$

that jointly minimize an energy functional.

2. Statement of results.

2.1. The energy functional. To state our results we need to introduce some notions from potential theory. Main references for potential theory in the complex plane are [8] and [9].

We will mainly work with finite positive measures on $\mathbb{C}$, but we will also use $\nu_1 - \nu_2$, where ν_1 and ν_2 are positive measures. The measures need not have bounded support. If ν has unbounded support, then we assume that

$$\int \log(1+|x|)\, \mathrm{d}\nu(x) < \infty. \tag{2.1}$$

In that case the logarithmic energy of ν is defined as

$$I(\nu) = \int \log \frac{1}{|x-y|}\, \mathrm{d}\nu(x)\mathrm{d}\nu(y) \tag{2.2}$$

and $I(\nu) \in (-\infty, +\infty]$.

DEFINITION 2.1. [illegible] $\mathcal{M}_e$ [illegible] ν [illegible] $\mathbb{C}$ [illegible] (2.1) [illegible] $I(\nu) < +\infty$ [illegible] $c > 0$ [illegible]

$$\mathcal{M}_e(c) = \{\nu \in \mathcal{M}_e \mid \nu(\mathbb{C}) = c\}. \tag{2.3}$$

The mutual energy $I(\nu_1, \nu_2)$ of two measures ν_1 and ν_2 is

$$I(\nu_1, \nu_2) = \int \log \frac{1}{|x-y|}\, \mathrm{d}\nu_1(x)\mathrm{d}\nu_2(y). \tag{2.4}$$

It is well defined and finite if $\nu_1, \nu_2 \in \mathcal{M}_e$, and in that case we have

$$I(\nu_1 - \nu_2) = I(\nu_1) + I(\nu_2) - 2I(\nu_1, \nu_2). \tag{2.5}$$

If $\nu_1, \nu_2 \in \mathcal{M}_e(c)$ for some $c > 0$, then

$$I(\nu_1 - \nu_2) \geq 0 \tag{2.6}$$

with equality if and only if $\nu_1 = \nu_2$. This is a well-known result if ν_1 and ν_2 have compact support [9]. For measures in $\mathcal{M}_e(c)$ with unbounded support, this is a recent result of Simeonov [11], who obtained this from a very elegant integral representation for $I(\nu_1 - \nu_2)$. It is a consequence of (2.6) that I is strictly convex on $\mathcal{M}_e(c)$, since

$$\begin{aligned} I\left(\frac{\nu_1+\nu_2}{2}\right) &= \frac{1}{2}\left(I(\nu_1) + I(\nu_2)\right) - I\left(\frac{\nu_1-\nu_2}{2}\right) \\ &\leq \frac{1}{2}\left(I(\nu_1) + I(\nu_2)\right) \qquad \text{for } \nu_1, \nu_2 \in \mathcal{M}_e(c) \end{aligned}$$

with equality if and only if $\nu_1 = \nu_2$.

Before we can state the equilibrium problem we also need to introduce the sets

$$\Gamma_k := \{\lambda \in \mathbb{C} \mid |z_{q+k}(\lambda)| = |z_{q+k+1}(\lambda)|\}, \qquad k = -q+1, \dots, p-1, \tag{2.7}$$

which for $k = 0$ reduce to the definition (1.10) of Γ_0. We will show that each Γ_k is the disjoint union of a finite number of open analytic arcs and a finite number of exceptional points. All Γ_k are unbounded, except for Γ_0 which is compact.

The equilibrium problem will be defined for a vector of measures denoted by $\vec{\nu} = (\nu_{-q+1}, \dots, \nu_{p-1})$. The component ν_k is a measure on Γ_k satisfying some additional properties that are given in the following definition.

DEFINITION 2.2. [illegible] $\vec{\nu} = (\nu_{-q+1}, \dots, \nu_{p-1})$ [illegible] $\nu_k \in \mathcal{M}_e$, ν_k [illegible] Γ_k [illegible]

$$\nu_k(\Gamma_k) = \begin{cases} \frac{q+k}{q} & \text{if } k \le 0, \\ \frac{p-k}{p} & \text{if } k \ge 0 \end{cases} \tag{2.8}$$

[illegible] $k = -q+1, \dots, p-1$

Now we are ready to state our first result. The proof is given in section 4.

THEOREM 2.3. [illegible] a [illegible] (1.3) [illegible] (1.4) [illegible] Γ_k [illegible] (2.7) [illegible] $k \in \{-q+1, \dots, p-1\}$ [illegible] μ_k [illegible] Γ_k [illegible]

$$\mathrm{d}\mu_k(\lambda) = \frac{1}{2\pi \mathrm{i}} \sum_{j=1}^{q+k} \left(\frac{z'_{j+}(\lambda)}{z_{j+}(\lambda)} - \frac{z'_{j-}(\lambda)}{z_{j-}(\lambda)} \right) \mathrm{d}\lambda, \tag{2.9}$$

[illegible] $\mathrm{d}\lambda$ [illegible] Γ_k [illegible] Γ_k [illegible] (1.12)) [illegible]

(a) $\vec{\mu} = (\mu_{-q+1}, \dots, \mu_{p-1})$ [illegible]

(b) [illegible] l_k [illegible]

$$\begin{aligned} 2 \int \log|\lambda - x| \, \mathrm{d}\mu_k(x) &= \int \log|\lambda - x| \, \mathrm{d}\mu_{k+1}(x) \\ &\quad + \int \log|\lambda - x| \, \mathrm{d}\mu_{k-1}(x) + l_k \end{aligned} \tag{2.10}$$

[illegible] $k = -q+1, \dots, p-1$ [illegible] $\lambda \in \Gamma_k$ [illegible] μ_{-q} [illegible] μ_p [illegible]

(c) $\vec{\mu} = (\mu_{-q+1}, \dots, \mu_{p-1})$ [illegible] J [illegible]

$$J(\vec{\nu}) = \sum_{k=-q+1}^{p-1} I(\nu_k) - \sum_{k=-q+1}^{p-2} I(\nu_k, \nu_{k+1}) \tag{2.11}$$

[illegible] $\vec{\nu} = (\nu_{-q+1}, \dots, \nu_{p-1})$

The relations (2.10) are the Euler–Lagrange variational conditions for the minimization problem for J among admissible vectors of measures.

It may not be obvious that the energy functional (2.11) is bounded from below. This can be seen from the alternative representation

$$\begin{aligned} J(\vec{\nu}) = &\left(\frac{1}{q} + \frac{1}{p} \right) I(\nu_0) + \sum_{k=1}^{q-1} k(k+1) \, I\left(\frac{\nu_{-q+k}}{k} - \frac{\nu_{-q+k+1}}{k+1} \right) \\ &+ \sum_{k=1}^{p-1} k(k+1) \, I\left(\frac{\nu_{p-k}}{k} - \frac{\nu_{p-k-1}}{k+1} \right). \end{aligned} \tag{2.12}$$

We leave the calculation leading to this identity to the reader. Under the normalizations (2.8) it follows by (2.6) that each term in the two finite sums on the right-hand side of (2.12) is nonnegative, so that

$$J(\vec{\nu}) \geq \left(\frac{1}{q} + \frac{1}{p}\right) I(\nu_0).$$

Since ν_0 is a Borel probability measure on Γ_0 and Γ_0 is compact, we indeed have that the energy functional is bounded from below on admissible vectors of measures $\vec{\nu}$.

The alternative representation (2.12) will play a role in the proof of Theorem 2.3.

Yet another representation for J is

$$J(\vec{\nu}) = \sum_{j,k=-q+1}^{p-1} A_{jk}\, I(\nu_j, \nu_k), \tag{2.13}$$

where the interaction matrix A has entries

$$A_{jk} = \begin{cases} 1 & \text{if } j = k, \\ -\frac{1}{2} & \text{if } |j-k| = 1, \\ 0 & \text{if } |j-k| \geq 2. \end{cases} \tag{2.14}$$

The energy functional in the form (2.13) and (2.14) also appears in the theory of simultaneous rational approximation, where it is the interaction matrix for a Nikishin system [7, Chapter 5].

It allows for the following physical interpretation: on each of the curves Γ_k one puts charged particles with total charge $(q+k)/q$ or $(p-k)/p$, depending on whether $k \leq 0$ or $k \geq 0$. Particles that lie on the same curve repel each other. The particles on two consecutive curves interact in the sense that they attract each other but in a way that is half as strong as the repulsion on a single curve. Particles on different curves that are not consecutive do not interact with each other in a direct way.

2.2. The measures μ_k as limiting measures of generalized eigenvalues. By (1.12) and Theorem 2.3 we know that the measure μ_0 that appears in the minimizer of the energy functional J is the limiting measure for the eigenvalues of $T_n(a)$. It is natural to ask about the other measures μ_k that appear in the minimizer. In our second result we show that the measures μ_k can be obtained as limiting counting measures for certain generalized eigenvalues.

Let $k \in \{-q+1, \ldots, p-1\}$. We use $T_n(z^{-k}(a-\lambda))$ to denote the Toeplitz matrix with the symbol $z \mapsto z^{-k}(a(z)-\lambda)$. For example, for $k=1$, $q=1$, and $p=2$, we have

$$T_n(z^{-k}(a-\lambda)) = \begin{pmatrix} a_1 & a_0-\lambda & a_{-1} & & & & \\ a_2 & a_1 & a_0-\lambda & a_{-1} & & & \\ & a_2 & a_1 & a_0-\lambda & a_{-1} & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & a_2 & a_1 & a_0-\lambda & a_{-1} \\ & & & & a_2 & a_1 & a_0-\lambda \\ & & & & & a_2 & a_1 \end{pmatrix}_{n\times n}.$$

Definition 2.4. [illegible] $k \in \{-q+1, \ldots, p-1\}$ [illegible] $n \geq 1$ [illegible] $P_{k,n}$ [illegible]

$$P_{k,n}(\lambda) = \det T_n(z^{-k}(a-\lambda)) \tag{2.15}$$

... k ... $T_n(a)$ ...

$$\operatorname{sp}_k T_n(a) = \{\lambda \in \mathbb{C} \mid P_{k,n}(\lambda) = 0\}. \tag{2.16}$$

... $\mu_{k,n}$ ... $\operatorname{sp}_k T_n(a)$

$$\mu_{k,n} = \frac{1}{n} \sum_{\lambda \in \operatorname{sp}_k T_n(a)} \delta_\lambda, \tag{2.17}$$

... λ ... $P_{k,n}$

Note that $\lambda \in \operatorname{sp}_k T_n(a)$ is a generalized eigenvalue (in the usual sense) for the matrix pencil $(T_n(z^{-k}a), T_n(z^{-k}))$, that is, $\det(A - \lambda B) = 0$ with $A = T_n(z^{-k}a)$ and $B = T_n(z^{-k})$. If $k = 0$, then $B = I$ and $\operatorname{sp}_0 T_n(a) = \operatorname{sp} T_n(a)$. If $k \neq 0$, then B is not invertible and the generalized eigenvalue problem is singular, causing there to be fewer than n generalized eigenvalues. In fact, since $T_n(z^{-k}(a-\lambda))$ has exactly $n - |k|$ entries $a_0 - \lambda$, we easily get that the degree of $P_{k,n}$ is at most $n - |k|$ and so there are at most $n - |k|$ generalized eigenvalues. Due to the band structure of $T_n(z^{-k}(a-\lambda))$, the actual number of generalized eigenvalues is substantially smaller.

PROPOSITION 2.5. ... $k \in \{-q+1, \dots, p-1\}$ ... $P_{k,n}(\lambda) = \gamma_{k,n}\lambda^{d_{k,n}} + \cdots$ ... $d_{k,n}$ ... $\gamma_{k,n} \neq 0$ ...

$$d_{k,n} \leq \begin{cases} \frac{q+k}{q} n & \text{... } k < 0, \\ \frac{p-k}{p} n & \text{... } k > 0. \end{cases} \tag{2.18}$$

... (2.18) ... $k > 0$ ... n ... p ... $k < 0$ ... n ... q ...

$$\gamma_{k,n} = \begin{cases} (-1)^{(k+1)n} a_{-q}^{|k|n/q} & \text{... } k < 0 \text{ ... } n \equiv 0 \bmod q, \\ (-1)^{(k+1)n} a_p^{kn/p} & \text{... } k > 0 \text{ ... } n \equiv 0 \bmod p. \end{cases} \tag{2.19}$$

We now come to our second main result. It is the analogue of the results of Schmidt, Spitzer, and Hirschman for the generalized eigenvalues.

THEOREM 2.6. ... $k \in \{-q+1, \dots, p-1\}$ ...

$$\liminf_{n\to\infty} \operatorname{sp}_k T_n(a) = \limsup_{n\to\infty} \operatorname{sp}_k T_n(a) = \Gamma_k, \tag{2.20}$$

...

$$\lim_{n\to\infty} \int_{\mathbb{C}} \phi(z)\, \mathrm{d}\mu_{k,n}(z) = \int_{\mathbb{C}} \phi(z)\, \mathrm{d}\mu_k(z) \tag{2.21}$$

... ϕ ... $\mathbb{C}$

The key element in the proof of Theorem 2.6 is a beautiful formula of Widom [14] (see [1, Theorem 2.8]) for the determinant of a banded Toeplitz matrix. In the present situation Widom's formula yields the following. Let $\lambda \in \mathbb{C}$ be such that the solutions $z_j(\lambda)$ of the algebraic equation (1.7) are mutually distinct. Then

$$P_{k,n}(\lambda) = \det T_n(z^{-k}(a-\lambda)) = \sum_M C_M(\lambda)\, (w_M(\lambda))^n, \tag{2.22}$$

where the sum is over all subsets $M \subset \{1, 2, \ldots, p+q\}$ of cardinality $|M| = p-k$ and for each such M we have

$$w_M(\lambda) := (-1)^{p-k} a_p \prod_{j \in M} z_j(\lambda) \tag{2.23}$$

and (with $\overline{M} := \{1, 2, \ldots, p+q\} \setminus M$)

$$C_M(\lambda) := \prod_{j \in M} z_j(\lambda)^{q+k} \prod_{\substack{j \in M \\ l \in \overline{M}}} (z_j(\lambda) - z_l(\lambda))^{-1}. \tag{2.24}$$

The formula (2.22) shows that for large n the main contribution comes from those M for which $|w_M(\lambda)|$ is the largest possible. For $\lambda \in \mathbb{C} \setminus \Gamma_k$ there is a unique such M, namely

$$M = M_k := \{q+k+1, q+k+2, \ldots, p+q\}, \tag{2.25}$$

because of the ordering (1.8).

2.3. Overview of the rest of the paper. In section 3 we will state some preliminary results about analyticity properties of the solutions z_j of the algebraic equation (1.7). These results will be needed in the proof of Theorem 2.3 which is given in section 4. In section 5 we will prove Proposition 2.5 and Theorem 2.6. Finally, we conclude the paper by giving some examples in section 6.

3. Preliminaries. In this section we collect a number of properties of the curves Γ_k and the solutions $z_1(\lambda), \ldots, z_{p+q}(\lambda)$ of the algebraic equation (1.7). For convenience we define throughout the rest of the paper

$$\Gamma_{-q} = \Gamma_p = \emptyset \qquad \text{and} \qquad \mu_{-q} = \mu_p = 0 \qquad \text{(the zero measure)}.$$

Occasionally, we also use

$$z_0(\lambda) = 0, \qquad z_{p+q+1}(\lambda) = +\infty.$$

3.1. The structure of the curves Γ_k. We start with a definition; cf. [1, section 11.2].

DEFINITION 3.1. [illegible] $\lambda_0 \in \mathbb{C}$ [illegible] $a(z) - \lambda_0 = 0$ [illegible] $\lambda_0 \in \Gamma_k$ [illegible] Γ_k [illegible] λ_0 [illegible] U [illegible] λ [illegible] $\Gamma_k \cap U$ [illegible] ∂U

If λ_0 is a branch point, then there is a z_0 such that $a(z_0) = \lambda_0$ and $a'(z_0) = 0$. Then we may assume that $z_0 = z_{q+k}(\lambda_0) = z_{q+k+1}(\lambda_0)$ for some k and $\lambda_0 \in \Gamma_k$. For a symbol a of the form (1.3), the derivative a' has exactly $p+q$ zeros (counted with multiplicity), so that there are exactly $p+q$ branch points counted with multiplicity.

The solutions $z_k(\lambda)$ also have branching at infinity (unless $p = 1$ or $q = 1$). There are p solutions of (1.7) that tend to infinity as $\lambda \to \infty$ and q solutions that tend to 0. Indeed, we have

$$z_k(\lambda) = \begin{cases} c_k \lambda^{-1/q}(1 + \mathcal{O}(\lambda^{-1/q})) & \text{for } k = 1, \ldots, q, \\ c_k \lambda^{1/p}(1 + \mathcal{O}(\lambda^{-1/p})) & \text{for } k = q+1, \ldots, p+q \end{cases} \tag{3.1}$$

as $\lambda \to \infty$. Here $c_1, \dots, c_q$ are the q distinct solutions of $c^q = a_{-q}$ (taken in some order depending on λ), and $c_{q+1}, \dots, c_{p+q}$ are the p distinct solutions of $c^p = a_p^{-1}$ (again taken in some order depending on λ).

The following proposition gives the structure of Γ_k at infinity.

PROPOSITION 3.2. [illegible] $k \in \{-q+1, \dots, p-1\} \setminus \{0\}$ [illegible] $R > 0$ [illegible] $\Gamma_k \cap \{\lambda \in \mathbb{C} \mid |\lambda| > R\}$ [illegible] $|\lambda| = R$ [illegible].

[illegible]. The proof is similar to the proof of [1, Proposition 11.8] where a similar structure theorem was proved for finite branch points. We omit the details. □

It follows from Proposition 3.2 that the exceptional points for Γ_k are in a bounded set. Since the set of exceptional points is discrete we conclude that there are only finitely many exceptional points. Then we have the following result about the structure of Γ_k.

PROPOSITION 3.3. [illegible] $k \in \{-q+1, \dots, p-1\}$ [illegible] Γ_k [illegible] Γ_k [illegible].

[illegible]. This was proved for $k = 0$ in [10] and [1, Theorem 11.9]. For general k, there are only finitely many exceptional points, and the proof follows in a similar way. □

3.2. The Riemann surface. From Proposition 3.3 it follows that the curves Γ_k can be taken as cuts for the $p+q$-sheeted Riemann surface of the algebraic equation (1.7). We number the sheets from 1 to $p+q$, where the kth sheet of the Riemann surface is

$$\mathcal{R}_k = \{\lambda \in \mathbb{C} \mid |z_{k-1}(\lambda)| < |z_k(\lambda)| < |z_{k+1}(\lambda)|\} = \mathbb{C} \setminus (\Gamma_{-q+k-1} \cup \Gamma_{-q+k}). \tag{3.2}$$

Thus z_k is well defined and analytic on $\mathcal{R}_k$.

The easiest case to visualize is the case where consecutive cuts are disjoint, that is, $\Gamma_{-q+k-1} \cap \Gamma_{-q+k} = \emptyset$ for every $k = 2, \dots, p+q-2$. In that case we have that $\mathcal{R}_k$ is connected to $\mathcal{R}_{k+1}$ via Γ_{-q+k} in the usual crosswise manner, and z_{k+1} is the analytic continuation of z_k across Γ_{-q+k}.

The general case is described in the following proposition.

PROPOSITION 3.4. [illegible] A [illegible] $A \subset \Gamma_{-q+k}$ [illegible] $k = k_1, \dots, k_2$ [illegible] $A \cap (\Gamma_{-q+k_1-1} \cup \Gamma_{-q+k_2+1}) = \emptyset$ [illegible] $k = k_1, \dots, k_2+1$ [illegible] z_k [illegible] A [illegible] $z_{k_1+k_2-k+1}$ [illegible] A [illegible] $\mathcal{R}_k$ [illegible] $\mathcal{R}_{k_1+k_2-k+1}$ [illegible].

[illegible]. We have that

$$|z_{k_1}(\lambda)| = |z_{k_1+1}(\lambda)| = \cdots = |z_{k_2}(\lambda)| = |z_{k_2+1}(\lambda)|$$

for $\lambda \in A$, with strict inequalities ($<$) for λ on either side of A. Choose an orientation for A. Then there is a permutation π of $\{k_1, \dots, k_2+1\}$ such that $z_{\pi(k)}$ is the analytic continuation of z_k from the $+$ side of A to the $-$ side of A.

Assume that there are $k, k' \in \{k_1, \dots, k_2+1\}$ such that $k < k'$ and $\pi(k) < \pi(k')$. Take a regular $\lambda_0 \in A$ and a small neighborhood U of λ_0 such that $A \cap U = \Gamma_{-q+k} \cap U = \Gamma_{-q+k'} \cap U$ and $A \cap U$ is an analytic arc starting and terminating on ∂U. Then we have a disjoint union $U = U_+ \cup U_- \cup (A \cap U)$, where U_+ (U_-) is the part of U on the $+$ side ($-$ side) of A. The function ϕ defined by

$$\phi(\lambda) = \begin{cases} \frac{z_k(\lambda)}{z_{k'}(\lambda)} & \text{for } \lambda \in U_+, \\ \frac{z_{\pi(k)}(\lambda)}{z_{\pi(k')}(\lambda)} & \text{for } \lambda \in U_- \end{cases}$$

has an analytic continuation to U and satisfies $|\phi(\lambda)| < 1$ for $\lambda \in U_+ \cup U_-$ and $|\phi(\lambda)| = 1$ for $\lambda \in A \cap U$. This contradicts the maximum principle for analytic functions. Therefore $\pi(k) > \pi(k')$ for every $k, k' \in \{k_1, \ldots, k_2 + 1\}$ with $k < k'$, and this implies that $\pi(k) = k_1 + k_2 - k + 1$ for every $k = k_1, \ldots, k_2 + 1$, and the proposition follows. □

3.3. The functions $w_k(\lambda)$. A major role is played by the functions w_k which, for $k \in \{-q+1, \ldots, p-1\}$, are defined by

$$w_k(\lambda) = \prod_{j=1}^{q+k} z_j(\lambda) \qquad \text{for } \lambda \in \mathbb{C} \setminus \Gamma_k. \tag{3.3}$$

Note that $w_k = (-1)^{p-k} a_p^{-1} w_{\{1,\ldots,k\}}$ in the notation of (2.23).

PROPOSITION 3.5. [illegible] w_k [illegible] $\mathbb{C} \setminus \Gamma_k$ [illegible]. Since z_j is analytic on $\mathcal{R}_j = \mathbb{C} \setminus (\Gamma_{-q+j-1} \cup \Gamma_{-q+j})$ (see (3.2)), we obtain from its definition that w_k is analytic in $\mathbb{C} \setminus \bigcup_{j=1}^{k+q} \Gamma_{-q+j}$. Let A be an analytic arc in $\Gamma_{-q+j} \setminus \Gamma_k$ for some $j < k + q$. Choose an orientation on A. Since the arc is disjoint from Γ_k, we have that $z_{j+}(\lambda) = z_{\pi(j)-}(\lambda)$, for $\lambda \in A$ and $j = 1, \ldots, q + k$, where π is a permutation of $\{1, \ldots, q+k\}$. Since w_k is symmetric in the z_j's for $j = 1, \ldots, q+k$, it then follows that

$$w_{k+}(\lambda) = w_{k-}(\lambda) \qquad \text{for } \lambda \in A,$$

which shows the analyticity in $\mathbb{C} \setminus \Gamma_k$ with the possible exception of isolated singularities at the exceptional points of Γ_{-q+1}, Γ_{-q+2}, …, Γ_{k-1}. However, each z_j, and therefore also w_k, is bounded near such an exceptional point, so that any isolated singularity is removable. □

In the rest of the paper we make frequent use of the logarithmic derivative w_k'/w_k of w_k. By the fact that w_k does not vanish on $\mathbb{C} \setminus \Gamma_k$ and by Proposition 3.5, it follows that w_k'/w_k is analytic in $\mathbb{C} \setminus \Gamma_k$. By Proposition 3.4 it, moreover, has an analytic continuation across every open analytic arc $A \subset \Gamma_k$. Near the exceptional points that are not branch points w_k'/w_k remains bounded. At the branch points it can, however, have singularities of a certain order.

PROPOSITION 3.6. [illegible] $\lambda_0 \in \Gamma_k$ [illegible] Γ_k [illegible] $m \in \mathbb{N}$ [illegible]

$$\frac{w_k'(\lambda)}{w_k(\lambda)} = \mathcal{O}\left((\lambda - \lambda_0)^{-m/(m+1)}\right) \tag{3.4}$$

[illegible] $\lambda \to \lambda_0$ [illegible] $\lambda \in \mathbb{C} \setminus \Gamma_k$ [illegible]. Let $1 \le j \le q + k$. We investigate the behavior of $z_j(\lambda)$ when $\lambda \to \lambda_0$ such that λ remains in a connected component of $\mathbb{C} \setminus (\Gamma_{j-1} \cup \Gamma_j)$. Then $z_j(\lambda) \to z_0$ for some $z_0 \in \mathbb{C}$ with $a(z_0) = \lambda_0$. Let $m_0 + 1$ be the multiplicity of z_0 as a solution of $a(z) = \lambda_0$. Then

$$a(z) = \lambda_0 + c_0(z - z_0)^{m_0+1}(1 + \mathcal{O}(z - z_0)), \qquad z \to z_0, \tag{3.5}$$

for some nonzero constant c_0. Therefore,

$$z_j(\lambda) = z_0 + \mathcal{O}((\lambda - \lambda_0)^{1/(m_0+1)}) \tag{3.6}$$

and

$$z_j'(\lambda) = \mathcal{O}((\lambda - \lambda_0)^{-m_0/(m_0+1)}) \tag{3.7}$$

for $\lambda \to \lambda_0$ such that λ remains in the same connected component of $\mathbb{C} \setminus (\Gamma_{j-1} \cup \Gamma_j)$. Let m be the maximum of all the multiplicities of the roots of $a(z) = \lambda_0$. Then it follows from (3.6) and (3.7) that

$$\frac{z_j'(\lambda)}{z_j(\lambda)} = \mathcal{O}((\lambda - \lambda_0)^{-m/(m+1)})$$

as $\lambda \to \lambda_0$ with $\lambda \in \mathbb{C} \setminus \Gamma_k$. Then we obtain (3.4) in view of (3.3). □

We end this section by giving the asymptotics of w_k'/w_k for $\lambda \to \infty$.

PROPOSITION 3.7. [illegible] $\lambda \to \infty$ [illegible] $\lambda \in \mathbb{C} \setminus \Gamma_k$ [illegible]

$$\frac{w_k'(\lambda)}{w_k(\lambda)} = \begin{cases} -\frac{q+k}{q}\lambda^{-1} + \mathcal{O}\left(\lambda^{-1-1/q}\right) & \text{[illegible]} \quad k = -q+1, \dots, -1, \\ -\lambda^{-1} + \mathcal{O}(\lambda^{-2}) & \text{[illegible]} \quad k = 0, \\ -\frac{p-k}{p}\lambda^{-1} + \mathcal{O}\left(\lambda^{-1-1/p}\right) & \text{[illegible]} \quad k = 1, \dots, p-1. \end{cases} \tag{3.8}$$

[illegible]. This follows directly from (3.1) and (3.3). □

4. Proof of Theorem 2.3. We use the function w_k introduced in (3.3). We define μ_k by the formula (2.9) and we note that

$$\mathrm{d}\mu_k(\lambda) = \frac{1}{2\pi \mathrm{i}} \left(\frac{w_{k+}'(\lambda)}{w_{k+}(\lambda)} - \frac{w_{k-}'(\lambda)}{w_{k-}(\lambda)} \right) \mathrm{d}\lambda. \tag{4.1}$$

PROPOSITION 4.1. [illegible] $k = -q+1, \dots, p-1$ [illegible] μ_k [illegible] Γ_k [illegible] $\mu_k(\Gamma_k) = (q+k)/q$, $k \geq 0$ [illegible] $\mu_k(\Gamma_k) = (p-k)/p$, $k \geq 0$ [illegible]. We first show that μ_k is a measure, i.e., that it is nonnegative on each analytic arc of Γ_k. Let A be an analytic arc in Γ_k consisting only of regular points. Let $t \mapsto \lambda(t)$ be a parametrization of A in the direction of the orientation of Γ_k. Then

$$\begin{aligned} \mathrm{d}\mu_k(\lambda) &= \frac{1}{2\pi \mathrm{i}} \left(\frac{w_{k+}'(\lambda(t))}{w_{k+}(\lambda(t))} - \frac{w_{k-}'(\lambda(t))}{w_{k-}(\lambda(t))} \right) \lambda'(t) \mathrm{d}t \\ &= \frac{1}{2\pi \mathrm{i}} \left(\frac{\mathrm{d}}{\mathrm{d}t} \log \frac{w_{k+}(\lambda(t))}{w_{k-}(\lambda(t))} \right) \mathrm{d}t. \end{aligned}$$

To conclude that μ_k is nonnegative on A, it is thus enough to show that

$$\operatorname{Re} \log \frac{w_{k+}(\lambda)}{w_{k-}(\lambda)} = 0 \qquad \text{for } \lambda \in A \tag{4.2}$$

and

$$\operatorname{Im} \log \frac{w_{k+}(\lambda)}{w_{k-}(\lambda)} \quad \text{increases along } A. \tag{4.3}$$

Since $|w_{k+}(\lambda)| = |w_{k-}(\lambda)|$ for $\lambda \in A$, we have (4.2) so that it only remains to prove (4.3).

There is a neighborhood U of A such that $U \setminus \Gamma_k$ has two components, denoted U_+ and U_-, where U_+ is on the $+$ side of Γ_k and U_- on the $-$ side. It follows from Proposition 3.4 that w_k has an analytic continuation from U_- to U, which we denote by $\hat{w}_k$, and that $|w_k(\lambda)| < |\hat{w}_k(\lambda)|$ for $\lambda \in U_+$, and equality $|w_{k+}(\lambda)| = |\hat{w}_k(\lambda)|$ holds for $\lambda \in A$. Thus it follows that

$$\frac{\partial}{\partial n} \operatorname{Re} \log \left(\frac{w_k(\lambda)}{\hat{w}_k(\lambda)} \right) \leq 0 \qquad \text{for } \lambda \in A,$$

where $\frac{\partial}{\partial n}$ denotes the normal derivative to A in the direction of U_+. Then by the Cauchy–Riemann equations we have that $\operatorname{Im}\log(\frac{w_{k+}(\lambda)}{\hat{w}_{k+}(\lambda)})$ is increasing along A. Since $\hat{w}_{k+}(\lambda) = w_{k-}(\lambda)$ for $\lambda \in A$, we obtain (4.3). Thus μ_k is a measure.

Next we show that μ_k is a finite measure, which means that we have to show that

$$\frac{w'_{k+}(\lambda)}{w_{k+}(\lambda)} - \frac{w'_{k-}(\lambda)}{w_{k-}(\lambda)} \tag{4.4}$$

is integrable near infinity on Γ_k and near every branch point on Γ_k. This follows from Propositions 3.7 and 3.6. Indeed, from Proposition 3.7 it follows that

$$\frac{w'_{k+}(\lambda)}{w_{k+}(\lambda)} - \frac{w'_{k-}(\lambda)}{w_{k-}(\lambda)} = \mathcal{O}\left(\lambda^{-1-\delta}\right) \qquad \text{as } \lambda \to \infty,\ \lambda \in \Gamma_k, \tag{4.5}$$

where $\delta = 1/q$ if $k < 0$ and $\delta = 1/p$ if $k > 0$. Since $\delta > 0$ we see that (4.4) is integrable near infinity. For a branch point λ_0 of Γ_k, we have from Proposition 3.6 that there exists an $m \geq 1$ such that

$$\frac{w'_{k+}(\lambda)}{w_{k+}(\lambda)} - \frac{w'_{k-}(\lambda)}{w_{k-}(\lambda)} = \mathcal{O}\left((\lambda-\lambda_0)^{-m/(m+1)}\right) \qquad \text{as } \lambda \to \lambda_0,\ \lambda \in \Gamma_k. \tag{4.6}$$

This shows that (4.4) is integrable near every branch point. Thus μ_k is a finite measure.

Finally we compute the total mass of μ_k. Let $D(0,R) = \{z \in \mathbb{C} \mid |z| < R\}$. Then for R large enough, so that $D(0,R)$ contains all exceptional points of Γ_k and all connected components of $\mathbb{C} \setminus \Gamma_k$ (if any),

$$\mu_k(\Gamma_k \cap D(0,R)) = \frac{1}{2\pi \mathrm{i}} \left(\int_{\Gamma_k \cap D(0,R)} \frac{w'_{k+}(\lambda)}{w_{k+}(\lambda)} \mathrm{d}\lambda - \int_{\Gamma_k \cap D(0,R)} \frac{w'_{k-}(\lambda)}{w_{k-}(\lambda)} \mathrm{d}\lambda \right), \tag{4.7}$$

where we have used the behavior (4.6) near the branch points in order to be able to split the integrals. Again using (4.6) we can then turn the two integrals into a contour integral over a contour $\tilde{\Gamma}_{k,R}$ as in Figure 4.1. The contour $\tilde{\Gamma}_{k,R}$ passes along the $\pm$ sides of $\Gamma_k \cap D(0,R)$, and if we choose the orientation that is also shown in Figure 4.1 (and which is independent of the choice of orientation for Γ_k), then

$$\mu_k(\Gamma_k \cap D(0,R)) = \frac{1}{2\pi \mathrm{i}} \int_{\tilde{\Gamma}_{k,R}} \frac{w'_k(\lambda)}{w_k(\lambda)} \mathrm{d}\lambda. \tag{4.8}$$

The parts of $\tilde{\Gamma}_{k,R}$ that belong to bounded components of $\mathbb{C} \setminus \Gamma_k$ form closed contours along the boundary of each bounded component. By Cauchy's theorem their contribution to the integral (4.8) vanishes. The parts of $\tilde{\Gamma}_{k,R}$ that belong to the unbounded components of $\mathbb{C} \setminus \Gamma_k$ can be deformed to the circle $\partial D(0,R)$ with the clockwise orientation. Thus if we use the positive orientation on $\partial D(0,R)$ as in Figure 4.1, then we obtain from (4.8)

$$\mu_k(\Gamma_k \cap D(0,R)) = -\frac{1}{2\pi \mathrm{i}} \oint_{\partial D(0,R)} \frac{w'_k(\lambda)}{w_k(\lambda)} \mathrm{d}\lambda.$$

Letting $R \to \infty$ and using Proposition 3.7, we then find that μ_k is a measure on Γ_k with total mass $\mu_k(\Gamma_k) = (q+k)/q$ if $k \leq 0$ and $\mu_k(\Gamma_k) = (p-k)/p$ if $k \geq 0$. □

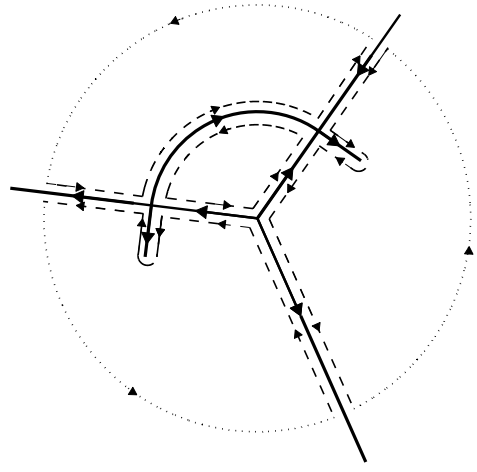


FIG. 4.1. *Illustration for the proofs of Propositions* 4.1 *and* 4.2. *The solid line is a sketch of a possible contour* Γ_k. *The dashed line is the contour* $\tilde{\Gamma}_{k,R}$, *and the dotted line is the boundary of a disk of radius* R *around* 0.

The following proposition is the next step in showing that the measures μ_k from (2.9) satisfy (2.10).

PROPOSITION 4.2. $k = -q+1, \ldots, p-1$

$$\int \frac{\mathrm{d}\mu_k(x)}{x-\lambda} = \frac{w_k'(\lambda)}{w_k(\lambda)} \qquad \lambda \in \mathbb{C} \setminus \Gamma_k \tag{4.9}$$

$$\int \log|\lambda - x| \, \mathrm{d}\mu_k(x) = -\log|w_k(\lambda)| + \alpha_k \qquad \lambda \in \mathbb{C}, \tag{4.10}$$

α_k

$$\alpha_k = \begin{cases} \log|a_{-q}| + \frac{k}{q}\log|a_{-q}| & k \le 0, \\ \log|a_{-q}| - \frac{k}{p}\log|a_p| & k \ge 0. \end{cases} \tag{4.11}$$

. To prove (4.9), we follow the same arguments as in the calculation of $\mu_k(\Gamma_k)$ in the end of the proof of Proposition 4.1. Let $\lambda \in \mathbb{C} \setminus \Gamma_k$, and choose $R > 0$ as in the proof of Proposition 4.1. We may assume $R > |\lambda|$. Then similar to (4.7) and (4.8) we can write

$$\int_{\Gamma_k \cap D(0,R)} \frac{\mathrm{d}\mu_k(x)}{x-\lambda} = \frac{1}{2\pi \mathrm{i}} \int_{\tilde{\Gamma}_{k,R}} \frac{w_k'(x)}{w_k(x)(x-\lambda)} \mathrm{d}x,$$

where $\tilde{\Gamma}_{k,R}$ has the same meaning as in the proof of Proposition 4.1; see also Figure 4.1. As in the proof of Proposition 4.1 we deform to an integral over $\partial D(0,R)$, but now we have to take into account that the integrand has a pole at $x = \lambda$ with residue $w_k'(\lambda)/w_k(\lambda)$. Therefore, by Cauchy's theorem

$$\int_{\Gamma_k \cap D(0,R)} \frac{\mathrm{d}\mu_k(x)}{x-\lambda} = \frac{w_k'(\lambda)}{w_k(\lambda)} - \frac{1}{2\pi \mathrm{i}} \int_{\partial D(0,R)} \frac{w_k'(x)}{w_k(x)(x-\lambda)} \mathrm{d}x. \tag{4.12}$$

Letting $R \to \infty$ and using Proposition 3.7 gives (4.9).

Next we integrate (4.9) over a Jordan curve J in $\mathbb{C} \setminus \Gamma_k$ from λ_1 to λ_2,

$$\begin{aligned}\int_{\lambda_1}^{\lambda_2} \int_{\Gamma_k} \frac{1}{x-\lambda}\, \mathrm{d}\mu_k(x)\, \mathrm{d}\lambda &= -\int \int_{\lambda_1}^{\lambda_2} \frac{1}{x-\lambda}\, \mathrm{d}\lambda\, \mathrm{d}\mu_k(x) \\ &= \int \left(\log|\lambda_1 - x| - \log|\lambda_2 - x| + \mathrm{i}\Delta_J[\arg(\lambda - x)]\right) \mathrm{d}\mu_k(x), \end{aligned} \tag{4.13}$$

where $\Delta_J[\arg(\lambda - x)]$ denotes the change in argument of $\lambda - x$ as when λ varies over J from λ_1 to λ_2. By (4.9) the integral (4.13) is equal to

$$\int_{\lambda_1}^{\lambda_2} \frac{w_k'(\lambda)}{w_k(\lambda)} \mathrm{d}\lambda = \log|w_k(\lambda_2)| - \log|w_k(\lambda_1)| + \mathrm{i}\Delta_J[\arg w_k(\lambda)]. \tag{4.14}$$

Equating the real parts of (4.13) and (4.14), we get

$$\int \left(\log|\lambda_1 - x| - \log|\lambda_2 - x|\right) \mathrm{d}\mu_k(x) = -\log|w_k(\lambda_1)| + \log|w_k(\lambda_2)|. \tag{4.15}$$

Since λ_1 and λ_2 can be taken arbitrarily in a connected component of $\mathbb{C} \setminus \Gamma_k$, we find that there exists a constant $\alpha_k \in \mathbb{R}$ (which a priori could depend on the connected component) such that

$$\int \log|\lambda - x|\, \mathrm{d}\mu_k(x) = -\log|w_k(\lambda)| + \alpha_k \tag{4.16}$$

for all λ in a connected component of $\mathbb{C} \setminus \Gamma_k$. By continuity, (4.16) extends to the closure of the connected component, which shows that the same constant α_k is valid for all connected components. Thus (4.16) holds for all $\lambda \in \mathbb{C}$.

The exact value of α_k can then be determined by expanding (4.16) for large λ. Suppose, for example, that $k < 0$. Then by (3.1) and (3.3)

$$|w_k(\lambda)| = \prod_{j=1}^{q+k} |z_j(\lambda)| = |a_{-q}|^{(q+k)/q} |\lambda|^{-(q+k)/q} \left(1 + \mathcal{O}(\lambda^{-1/q})\right)$$

as $\lambda \to \infty$. Thus

$$-\log|w_k(\lambda)| = \frac{q+k}{q} \log|\lambda| - \frac{q+k}{q} \log|a_{-q}| + \mathcal{O}(\lambda^{-1/q}). \tag{4.17}$$

Since

$$\int \log|\lambda - x|\, \mathrm{d}\mu_k(x) = \log|\lambda| \mu_k(\Gamma_k) + o(1) = \frac{q+k}{q} \log|\lambda| + o(1) \qquad \text{as } \lambda \to \infty, \tag{4.18}$$

the value (4.11) for α_k follows from (4.16), (4.17), and (4.18). The argument for $k > 0$ is similar. This completes the proof of the proposition. □

To prove part (c) of Theorem 2.3 we also need the following lemma.

LEMMA 4.3. [illegible] $\vec{\nu}_1 = (\nu_{1,-q+1} \ldots, \nu_{1,p-1})$ [illegible] $\vec{\nu}_2 = (\nu_{2,-q+1} \ldots, \nu_{2,p-1})$ [illegible] $J(\vec{\nu}_1 - \vec{\nu}_2)$ [illegible]

$$J(\vec{\nu}_1 - \vec{\nu}_2) \geq 0 \tag{4.19}$$

[illegible] $\vec{\nu}_1 = \vec{\nu}_2$

. Since both $\vec{\nu}_1$ and $\vec{\nu}_2$ have finite energy, we find that $J(\vec{\nu}_1 - \vec{\nu}_2)$ is well defined. According to the alternative representation (2.12), we have

$$\begin{aligned} J(\vec{\nu}_1 - \vec{\nu}_2) = &\left(\frac{1}{q} + \frac{1}{p}\right) I(\nu_{1,0} - \nu_{2,0}) \\ &+ \sum_{k=1}^{q-1} k(k+1) I\left(\frac{\nu_{1,-q+k}}{k} - \frac{\nu_{2,-q+k}}{k} - \frac{\nu_{1,-q+k+1}}{k+1} + \frac{\nu_{2,-q+k+1}}{k+1}\right) \\ &+ \sum_{k=1}^{p-1} k(k+1) I\left(\frac{\nu_{1,p-k}}{k} - \frac{\nu_{2,p-k}}{k} - \frac{\nu_{1,p-k-1}}{k+1} + \frac{\nu_{2,p-k-1}}{k+1}\right). \end{aligned} \tag{4.20}$$

Using (2.6) and (2.8), we see that all terms in (4.20) are nonnegative and therefore (4.19) holds.

Suppose now that $J(\vec{\nu}_1 - \vec{\nu}_2) = 0$. Then all terms in the right-hand side of (4.20) are zero, so that

$$\nu_{1,0} = \nu_{2,0}, \tag{4.21}$$

$$\frac{\nu_{1,-q+k}}{k} + \frac{\nu_{2,-q+k+1}}{k+1} = \frac{\nu_{1,-q+k+1}}{k+1} + \frac{\nu_{2,-q+k}}{k} \qquad \text{for } k = 1, \dots, q-1, \tag{4.22}$$

$$\frac{\nu_{1,p-k}}{k} + \frac{\nu_{2,p-k-1}}{k+1} = \frac{\nu_{1,p-k-1}}{k+1} + \frac{\nu_{2,p-k}}{k} \qquad \text{for } k = 1, \dots, p-1. \tag{4.23}$$

Using (4.21) in (4.22) with $k = q-1$, we find $\nu_{1,-1} = \nu_{2,-1}$. Proceeding inductively we then obtain from (4.22) that $\nu_{1,k} = \nu_{2,k}$ for all $k = -q+1, \dots, 0$. Similarly, from (4.21) and (4.23) it follows that $\nu_{1,k} = \nu_{2,k}$ for $k = 0, \dots, p-1$, so that $\vec{\nu}_1 = \vec{\nu}_2$ as claimed. □

Now we are ready for the proof of Theorem 2.3.

2.3 (a) In view of Proposition 4.1 it remains to show only that $\mu_k \in \mathcal{M}_e$ for every $k = -q+1, \dots, p-1$. The decay estimate (4.5) implies that

$$\int \log(1 + |\lambda|)\, \mathrm{d}\mu_k(\lambda) < \infty.$$

The fact that $I(\mu_k) < +\infty$ follows from (4.10). Indeed,

$$I(\mu_k) = -\iint \log|\lambda - x| \mathrm{d}\mu_k(x) \mathrm{d}\mu_k(\lambda) = \int (\log|w_k(\lambda)| - \alpha_k) \mathrm{d}\mu_k(\lambda),$$

and this is finite since μ_k is a finite measure on Γ_k with a density that decays as in (4.5) and $\log|w_k(\lambda)|$ is continuous on Γ_k and grows only as a constant times $\log|\lambda|$ as $\lambda \to \infty$. Thus $\vec{\mu}$ is admissible, and part (a) is proved.

(b) According to (4.10) we have

$$\begin{aligned} &2\int \log|\lambda - x|\ \mathrm{d}\mu_k(x) - \int \log|\lambda - x|\ \mathrm{d}\mu_{k+1}(\lambda) - \int \log|\lambda - x|\ \mathrm{d}\mu_{k-1}(\lambda) \\ &\quad = -2\log|w_k(\lambda)| + 2\alpha_k + \log|w_{k+1}(\lambda)| - \alpha_{k+1} + \log|w_{k-1}(\lambda)| - \alpha_{k-1} \\ &\quad = \log\left|\frac{w_{k+1}(\lambda) w_{k-1}(\lambda)}{w_k(\lambda)^2}\right| + 2\alpha_k - \alpha_{k+1} - \alpha_{k-1} \\ &\quad = \log\left|\frac{z_{q+k+1}(\lambda)}{z_{q+k}(\lambda)}\right| + 2\alpha_k - \alpha_{k+1} - \alpha_{k-1}. \end{aligned} \tag{4.24}$$

Since $|z_{q+k}(\lambda)| = |z_{q+k+1}(\lambda)|$ for $\lambda \in \Gamma_k$, we see from (4.24) that (2.10) holds with constant

$$l_k = 2\alpha_k - \alpha_{k-1} + \alpha_{k+1}. \tag{4.25}$$

Note that for $k = -q+1$ and $k = p-1$ we are using the convention that $\mu_{-q} = \mu_p = 0$, and we also have put $\alpha_{-q} = \alpha_p = 0$. This proves part (b).

(c) Let $\vec{\nu} = (\nu_{-q+1}, \ldots, \nu_{p-1})$ be any admissible vector of measures. From the representation (2.13) we get

$$\begin{aligned} J(\vec{\nu}) &= J(\vec{\mu} + \vec{\nu} - \vec{\mu}) \\ &= J(\vec{\mu}) + J(\vec{\nu} - \vec{\mu}) + 2 \sum_{j,k=-q+1}^{p-1} A_{jk} I(\mu_j, \nu_k - \mu_k). \end{aligned} \tag{4.26}$$

Using (2.14), we find from (4.26) that

$$J(\vec{\nu}) = J(\vec{\mu}) + J(\vec{\nu} - \vec{\mu}) + \sum_{k=-q+1}^{p-1} I(2\mu_k - \mu_{k-1} - \mu_{k+1}, \nu_k - \mu_k). \tag{4.27}$$

For each $k = -q+1, \ldots, p-1$, we have

$$\begin{aligned} &I(2\mu_k - \mu_{k-1} - \mu_{k+1}, \nu_k - \mu_k) \\ &\quad = \int \left(\int \log|\lambda - x| \, \mathrm{d}(2\mu_k - \mu_{k-1} - \mu_{k+1})(x) \right) d(\nu_k - \mu_k)(\lambda). \end{aligned} \tag{4.28}$$

By (2.10) the inner integral in the right-hand side of (4.28) is constant for $\lambda \in \Gamma_k$. Since ν_k and μ_k are finite measures on Γ_k with $\nu_k(\Gamma_k) = \mu_k(\Gamma_k)$, we find from (4.28) that

$$I(2\mu_k - \mu_{k-1} - \mu_{k+1}, \nu_k - \mu_k) = 0 \qquad \text{for } k = -q+1, \ldots, p-1.$$

Then (4.27) shows that $J(\vec{\nu}) = J(\vec{\mu}) + J(\vec{\nu} - \vec{\mu})$, which by Lemma 4.3 implies that $J(\vec{\nu}) \geq J(\vec{\mu})$ and equality holds if and only if $\vec{\nu} = \vec{\mu}$. This completes the proof of Theorem 2.3. □

5. Proofs of Proposition 2.5 and Theorem 2.6.

5.1. Proof of Proposition 2.5. We will now prove Proposition 2.5, which follows by a combinatorial argument.

[illegible] 2.5 We prove (2.18) and (2.19) for $k > 0$. The case $k < 0$ is similar. Let us first expand the determinant in the definition of $P_{k,n}$,

$$P_{k,n}(\lambda) = \det T_n(z^{-k}(a - \lambda)) = \sum_{\pi \in S_n} \prod_{j=1}^{n} (a - \lambda)_{j - \pi(j) + k}. \tag{5.1}$$

Here S_n denotes the set of all permutation on $\{1, \ldots, n\}$. By the band structure of $T_n(z^{-k}(a - \lambda))$ it follows that we have only nonzero contributions from permutations π that satisfy

$$k - p \leq \pi(j) - j \leq q + k \qquad \text{for all } j = 1, \ldots, n. \tag{5.2}$$

Define, for $\pi \in S_n$,

$$N_\pi = \{j \mid \pi(j) = j + k\} \tag{5.3}$$

and denote the number of elements of N_π by $|N_\pi|$. For each $\pi \in S_n$ we have that $\prod_{j=1}^n (a-\lambda)_{j-\pi(j)+k}$ is a polynomial in λ of degree at most $|N_\pi|$. So by (5.1)

$$d_{k,n} = \deg P_{k,n} \le \max_\pi |N_\pi|, \tag{5.4}$$

where we maximize over permutations $\pi \in S_n$ satisfying (5.2).

Let $\pi \in S_n$ satisfy (5.2). We prove (2.18) by giving an upper bound for $|N_\pi|$. Since $\sum_{j=1}^n (\pi(j) - j) = 0$ we obtain

$$\sum_{j=1}^n (\pi(j) - j)_+ = \sum_{j=1}^n (j - \pi(j))_+, \tag{5.5}$$

where $(\cdot)_+$ is defined as $(a)_+ = \max(0, a)$ for $a \in \mathbb{R}$. Each $j \in N_\pi$ gives a contribution k to the left-hand side of (5.5). Therefore the left-hand side is at least $k|N_\pi|$. By (5.2) we have that each term in the right-hand side is at most $p - k$. Moreover, there are at most $n - |N_\pi|$ nonzero terms in this sum. Combining this with (5.5) leads to

$$k|N_\pi| \le \sum_{j=1}^n (\pi(j) - j)_+ = \sum_{j=1}^n (j - \pi(j))_+ \le (n - |N_\pi|)(p - k). \tag{5.6}$$

Hence, if π is a permutation satisfying (5.2), then

$$|N_\pi| \le \frac{n(p-k)}{p}. \tag{5.7}$$

Now (2.18) follows by combining (5.7) and (5.4).

To prove (2.19) we assume that $n \equiv 0 \bmod p$. We claim that there exists a unique π such that equality holds in (5.7). Then equality holds in both inequalities of (5.6), and the above arguments show that this can happen only if

$$\pi(j) = j + k \qquad \text{or} \qquad \pi(j) = j - p + k \tag{5.8}$$

for every $j = 1, \ldots, n$. We claim that there exists a unique such permutation, namely

$$\pi(j) = \begin{cases} j + k & \text{if } j \equiv 1, \ldots, (p-k) \bmod p, \\ j - p + k & \text{if } j \equiv (p-k+1), \ldots, p \bmod p. \end{cases} \tag{5.9}$$

To see this let π be a permutation satisfying (5.8). The numbers $1, \ldots, p-k$ cannot satisfy $\pi(j) = j - p + k$ and thus satisfy $\pi(j) = j + k$. On the other hand, the numbers $1, \ldots, k$ cannot be the image of numbers j satisfying $\pi(j) = j + k$, and thus $\pi(j) = j - p + k$ for $j = p - k + 1, \ldots, p$. So (5.9) holds for $j = 1, \ldots, p$. This means in particular that the restriction of π to $\{p+1, \ldots, n\}$ is again a permutation, but now on $\{p+1, \ldots, n\}$. By the same arguments we then find that (5.9) holds for $j = p+1, \ldots, 2p$, and so on. The result is that (5.9) is indeed the only permutation that satisfies (5.8).

Finally, a straightforward calculation shows that the coefficient of $\lambda^{(p-k)n/p}$ in $\prod_{j=1}^n (a-\lambda)_{j-\pi(j)+k}$ with π as in (5.9) is nonzero and given by (2.19). This proves the proposition. □

5.2. Proof of Theorem 2.6. Before we start with the proof of Theorem 2.6 we first prove the following proposition concerning the asymptotics for $P_{k,n}$ for $n \to \infty$.

Proposition 5.1. [illegible] $M_k = \{q+k+1, \dots, p+q\}$ [illegible]

$$P_{k,n}(\lambda) = (w_{M_k}(\lambda))^n C_{M_k}(\lambda)\left(1 + \mathcal{O}(\exp(-c_K n))\right), \qquad n \to \infty, \tag{5.10}$$

[illegible] K [illegible] $\mathbb{C} \setminus \Gamma_k$ [illegible] c_K [illegible] K

[illegible] First rewrite (2.22) as

$$P_{k,n}(\lambda) = (w_{M_k}(\lambda))^n C_{M_k}(\lambda)\left(1 + R_{k,n}(\lambda)\right) \tag{5.11}$$

with $R_{k,n}$ defined by

$$R_{k,n}(\lambda) = \sum_{M \neq M_k} \frac{(w_M(\lambda))^n C_M(\lambda)}{(w_{M_k}(\lambda))^n C_{M_k}(\lambda)}. \tag{5.12}$$

Let K be a compact subset of $\mathbb{C} \setminus \Gamma_k$. If K does not contain branch points, then there exist $A, B > 0$ such that

$$A < |C_M(\lambda)| < B \tag{5.13}$$

for all $\lambda \in K$ and M. Moreover, we have

$$\left|\frac{w_M(\lambda)}{w_{M_k}(\lambda)}\right| \leq \left|\frac{z_{q+k}(\lambda)}{z_{q+k+1}(\lambda)}\right| \leq \sup_{\lambda \in K}\left|\frac{z_{q+k}(\lambda)}{z_{q+k+1}(\lambda)}\right| < 1 \tag{5.14}$$

for all $\lambda \in K$ and $M \neq M_k$. Therefore one readily verifies from (5.11) that there exist c_K such that $|R_{k,n}(\lambda)| \leq \exp(-c_K n)$ for all $\lambda \in K$ and n large enough. This proves the statement in case K does not contain branch points.

Suppose that K does contain branch points. Without loss of generality, we can assume that all branch points lie in the interior of K (otherwise we replace K by a bigger compact set). The boundary ∂K of K is a compact set with no branch points, and therefore (5.10) holds for ∂K by the above arguments. Since w_{M_k} and C_{M_k} are analytic in K, we find by (5.11) that $R_{k,n}$ is analytic in K. The maximum modulus principle for analytic functions states that $\sup_{z \in K} |R_{k,n}(z)| = \sup_{z \in \partial K} |R_{k,n}(z)|$, and thereby we obtain that (5.10) also holds for K with the same constant $c_K = c_{\partial K}$. □

We now state two particular consequences of (5.10).

Corollary 5.2. [illegible] $k \in \{-q+1, \dots, p-1\}$ [illegible] $K \subset \mathbb{C} \setminus \Gamma_k$ [illegible] $\mu_{k,n}(K) = 0$ [illegible] n [illegible]

[illegible] Let K be a compact subset of $\mathbb{C} \setminus \Gamma_k$. By (5.10) it follows that $P_{k,n}$ has no zeros in K for large n. Since $n\mu_{k,n}(K)$ equals the number of zeros of $P_{k,n}$ in K the corollary follows. □

Corollary 5.3. [illegible] $k \in \{-q+1, \dots, p-1\}$ [illegible]

$$\lim_{n \to \infty} \int_{\mathbb{C}} \frac{\mathrm{d}\mu_{k,n}(x)}{x - \lambda} = \int_{\Gamma_k} \frac{\mathrm{d}\mu_k(x)}{x - \lambda} \tag{5.15}$$

[illegible] $\mathbb{C} \setminus \Gamma_k$

[illegible] Let K be a compact subset of $\mathbb{C} \setminus \Gamma_k$. Note that

$$\int \frac{\mathrm{d}\mu_{k,n}(x)}{x - \lambda} = \frac{1}{n} \sum_{\lambda_i \in \mathrm{sp}_k T_n(a)} \frac{1}{\lambda_i - \lambda} = -\frac{P_{k,n}'(\lambda)}{nP_{k,n}(\lambda)} \tag{5.16}$$

for all $\lambda \in K$. With M_k and c_K as in Proposition 5.1 we obtain from (5.10) that

$$\frac{P'_{k,n}(\lambda)}{nP_{k,n}(\lambda)} = \frac{w'_{M_k}(\lambda)}{w_{M_k}(\lambda)} + \mathcal{O}(1/n), \qquad n \to \infty, \tag{5.17}$$

uniformly on K. Let us rewrite the right-hand side of (5.17). By expanding both sides of $z^q(a(z) - \lambda) = a_p \prod_{j=1}^{p+q}(z - z_j(\lambda))$ and collecting the constant terms we obtain

$$\prod_{j=1}^{p+q}(-z_j(\lambda)) = \frac{a_{-q}}{a_p}. \tag{5.18}$$

Since $\lambda \notin \Gamma_k$, we can split this product into two parts, take the logarithmic derivative, and use (3.3) and (2.23) to obtain

$$0 = \sum_{j=1}^{q+k} \frac{z'_j(\lambda)}{z_j(\lambda)} + \sum_{j=q+k+1}^{p+q} \frac{z'_j(\lambda)}{z_j(\lambda)} = \frac{w'_k(\lambda)}{w_k(\lambda)} + \frac{w'_{M_k}(\lambda)}{w_{M_k}(\lambda)}. \tag{5.19}$$

Combining (5.16), (5.17), and (5.19), we obtain

$$\lim_{n\to\infty} \int \frac{\mathrm{d}\mu_{k,n}(x)}{x - \lambda} = \frac{w'_k(\lambda)}{w_k(\lambda)} \tag{5.20}$$

uniformly on K. Then (5.15) follows from (5.20) and (4.9). □

Now we are ready for the proof of Theorem 2.6.

Proof of Theorem 2.6 First we prove (2.21). By Proposition 2.5 and the fact that $\vec{\mu}$ is admissible, we get (see (2.8))

$$\mu_{k,n}(\mathbb{C}) = \frac{1}{n} \deg P_{k,n} \le \mu_k(\mathbb{C}) \tag{5.21}$$

for every $n \in \mathbb{N}$.

Let $C_0(\mathbb{C})$ be the Banach space of continuous functions on $\mathbb{C}$ that vanish at infinity. The dual space $C_0(\mathbb{C})^*$ of $C_0(\mathbb{C})$ is the space of regular complex Borel measures on $\mathbb{C}$. By (5.21) the sequence $(\mu_{k,n})_{n\in\mathbb{N}}$ belongs to the ball in $C_0(\mathbb{C})^*$ centered at the origin with radius $\mu_k(\mathbb{C})$, which is weak* compact by the Banach–Alaoglu theorem. Let $\mu_{k,\infty}$ be the limit of a weak* convergent subsequence of $(\mu_{k,n})_{n\in\mathbb{N}}$.

By weak* convergence and Corollary 5.2 we obtain that $\mu_{k,\infty}$ is supported on Γ_k. Combining this with (5.15) and the weak* convergence leads to

$$\frac{1}{2\pi\mathrm{i}} \int_{\Gamma_k} \frac{\mathrm{d}\mu_k(x)}{x - \lambda} = \frac{1}{2\pi\mathrm{i}} \int_{\Gamma_k} \frac{\mathrm{d}\mu_{k,\infty}(x)}{x - \lambda} \tag{5.22}$$

for every $\lambda \in \mathbb{C} \setminus \Gamma_k$. The integrals in (5.22) are known in the literature as the Cauchy transforms of the measures μ_k and $\mu_{k,\infty}$. The Cauchy transform on Γ_k is an injective map that maps measures on Γ_k to functions that are analytic in $\mathbb{C} \setminus \Gamma_k$ (one can find explicit inversion formulae; see, for example, the arguments in [9, Theorem II.1.4] or the Stieltjes–Perron inversion formula in the special case $\Gamma_k \subset \mathbb{R}$). Thus it follows from (5.22) that $\mu_{k,\infty} = \mu_k$. Therefore

$$\lim_{n\to\infty} \mu_{k,n} = \mu_k \tag{5.23}$$

in the sense of weak* convergence in $C_0(\mathbb{C})^*$. Thus (2.21) holds if ϕ is a continuous function that vanishes at infinity.

From (5.21) and (5.23) it also follows that

$$\lim_{n\to\infty} \mu_{k,n}(\mathbb{C}) = \mu_k(\mathbb{C}). \tag{5.24}$$

Then the sequence $(\mu_{k,n})_{n\in\mathbb{N}}$ is tight. That is, for every $\varepsilon > 0$ there exists a compact K such that $\mu_{k,n}(\mathbb{C}\setminus K) < \varepsilon$ for every $n \in \mathbb{N}$. By a standard approximation argument one can now show that (2.21) holds for every bounded continuous function ϕ on $\mathbb{C}$.

Having (2.21) and Proposition 5.1, we can prove (2.20) as in [1, Theorem 11.17]. Indeed, the sets $\liminf_{n\to\infty} \operatorname{sp}_k T_n(a)$ and $\limsup_{n\to\infty} \operatorname{sp}_k T_n(a)$ equal the support of μ_k, which is Γ_k. □

6. Examples.

6.1. Example 1. As a first example, consider the symbol a defined by

$$a(z) = \frac{4(z+1)^3}{27z}. \tag{6.1}$$

In this case we have $p = 2$ and $q = 1$. So we obtain two contours Γ_0 and Γ_1 with two associated measures μ_0 and μ_1. This example appeared in [3], in which the authors gave explicit expressions for Γ_0 and μ_0. The following proposition also contains expressions for Γ_1 and μ_1. In what follows we take the principal branches for all fractional powers.

PROPOSITION 6.1. [illegible] a [illegible] (6.1) [illegible] $\Gamma_0 = [0,1]$ [illegible]

$$\mathrm{d}\mu_0(\lambda) = \frac{\sqrt{3}}{4\pi} \frac{\left(1+\sqrt{1-\lambda}\right)^{1/3} + \left(1-\sqrt{1-\lambda}\right)^{1/3}}{\lambda^{2/3}\sqrt{1-\lambda}}\, \mathrm{d}\lambda. \tag{6.2}$$

[illegible] $\Gamma_1 = (-\infty, 0]$ [illegible]

$$\mathrm{d}\mu_1(\lambda) = \frac{\sqrt{3}}{4\pi} \frac{\left(1+\sqrt{1-\lambda}\right)^{1/3} - \left(\sqrt{1-\lambda}-1\right)^{1/3}}{(-\lambda)^{2/3}\sqrt{1-\lambda}}\, \mathrm{d}\lambda. \tag{6.3}$$

[illegible]. A straightforward calculation shows that $\lambda = 0$ and $\lambda = 1$ are the branch points.

Let $\lambda \in \Gamma_0 \cup \Gamma_1$ and assume that λ is not a branch point. There exist $y_1, y_2 \in \mathbb{C}$ such that $y_1 \neq y_2$, $|y_1| = |y_2|$, and $a(y_1) = a(y_2) = \lambda$. Then it follows from (6.1) that $|y_1 + 1| = |y_2 + 1|$. Therefore y_1 and y_2 are intersection points of a circle centered at -1 and a circle centered at the origin. Since $y_1 \neq y_2$, this means that $y_1 = \overline{y_2}$ and therefore $\lambda = a(y_1) = a(\overline{y_2}) = \overline{a(y_1)} = \overline{\lambda}$, so that $\lambda \in \mathbb{R}$. A further investigation shows that $a(z) - \lambda$ has three different real zeros if $\lambda > 1$. If $\lambda < 1$ and $\lambda \neq 0$, then $a(z) - \lambda$ has precisely one real zero and two conjugate complex zeros. Therefore, $\Gamma_0 \cup \Gamma_1 = (-\infty, 1]$.

Now we will show that $\Gamma_0 = [0,1]$ and $\Gamma_1 = (-\infty, 0]$. By Cardano's formula the solutions of the algebraic equation $a(z) = \lambda$ are given by

$$z_j(\lambda) = -1 - \frac{3\lambda^{1/3}}{2}\left(\omega^j\left(1+(1-\lambda)^{1/2}\right)^{1/3} + \omega^{-j}\left(1-(1-\lambda)^{1/2}\right)^{1/3}\right) \tag{6.4}$$

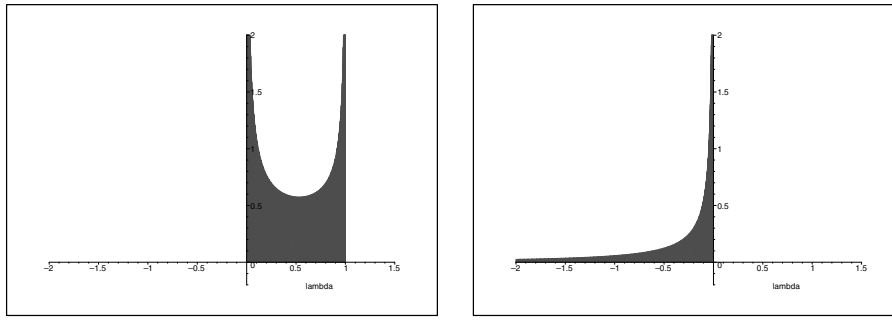


FIG. 6.1. *Illustration for Example* 1: *The densities of the measures* μ_0 *(left) and* μ_1 *(right) for* $a = \frac{4(z+1)^3}{27z}$.

for $\lambda \in [0, 1]$ and

(6.5)
$$z_j(\lambda) = -1 + \frac{3(-\lambda)^{1/3}}{2}\left(\omega^{j+2}\left(1 + (1-\lambda)^{1/2}\right)^{1/3} - \omega^{-j-2}\left((1-\lambda)^{1/2} - 1\right)^{1/3}\right)$$

for $\lambda \in (-\infty, 0]$. Here $\omega = \mathrm{e}^{2\pi \mathrm{i}/3}$. One can check that $|z_1(\lambda)| = |z_2(\lambda)| < |z_3(\lambda)|$ for $\lambda \in (0, 1]$ and $|z_1(\lambda)| < |z_2(\lambda)| = |z_3(\lambda)|$ for $\lambda \in (-\infty, 0)$. Moreover, for $\lambda = 0$ we have $z_1(0) = z_2(0) = z_3(0) = -1$. Therefore $\Gamma_0 = [0, 1]$ and $\Gamma_1 = (-\infty, 0]$.

The density (6.2) was already given in [3], and (6.3) follows in a similar way. ∎

In Figure 6.1 we plot the densities of μ_0 and μ_1. Note that, due to the interaction between μ_0 and μ_1 in the energy functional, there is more mass of μ_0 near 0 than near 1. We also see that the singularities of the densities for μ_0 and μ_1 are of order $\mathcal{O}(|\lambda|^{-2/3})$ for $\lambda \to 0$, whereas the typical nature of a singularity in each of the measures is a square root singularity. The stronger singularity is due to the fact that $a(z) - \lambda$ has a triple root for $\lambda = 0$.

In Figure 6.2 we plot the eigenvalues and generalized eigenvalues for $n = 50$. It is known that the eigenvalues are simple and positive [3, section 2.3], which we also see in Figure 6.2.

6.2. Example 2. For the symbol a defined by

(6.6)
$$a(z) = z^2 + z + z^{-1} + z^{-2},$$

we have $p = q = 2$. From the symmetry $a(1/z) = a(z)$ it follows that $\Gamma_{-1} = \Gamma_1$ and $\mu_{-1} = \mu_1$.

The interesting feature of this example is that the contours Γ_0 and $\Gamma_{\pm 1}$ overlap. To be precise, the interval $(-9/4, 0)$ is contained in all three contours Γ_{-1}, Γ_0, and Γ_1. This can be most easily seen by investigating the image of the unit circle under a. Consider

(6.7)
$$a(\mathrm{e}^{\mathrm{i}t}) = 2\cos 2t + 2\cos t \qquad \text{for } t \in [0, 2\pi).$$

A straightforward analysis shows that for every $\lambda \in (-9/4, 0)$ the equation $a(\mathrm{e}^{\mathrm{i}t}) = \lambda$ has four different solutions for t in $[0, 2\pi)$. This means that the four solutions of the equation $a(z) = \lambda$ are on the unit circle, and so in particular have the same absolute value.

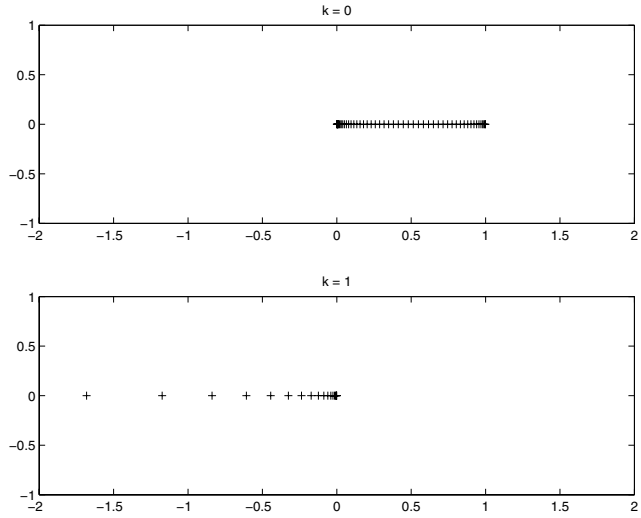


FIG. 6.2. *Illustration for Example* 1: *The spectrum* $\operatorname{sp} T_{50}(a)$ *(top) and the generalized spectrum* $\operatorname{sp}_1 T_{50}(a)$ *(bottom) for the symbol* $a = \frac{4(z+1)^3}{27z}$.

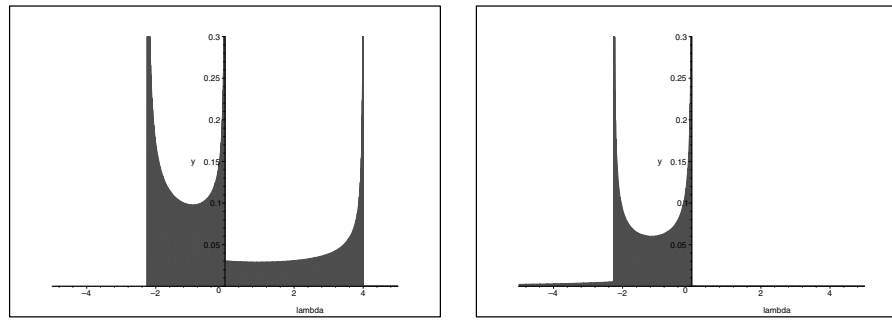


FIG. 6.3. *Illustration for Example* 2: *The densities of the measures* μ_0 *(left) and* $\mu_1 = \mu_{-1}$ *(right) for* $a(z) = z^2 + z + z^{-1} + z^{-2}$.

The equation $a(z) - \lambda = 0$ can be explicitly solved by introducing the variable $y = z + 1/z$. In exactly the same way as in the previous example one can obtain the limiting measures. We will not give the explicit formulae, but only plot the densities in Figure 6.3. The branch points are $\lambda = -9/4$, $\lambda = 0$, and $\lambda = 4$. The contours are given by

$$\Gamma_0 = [-9/4, 4], \qquad \Gamma_{-1} = \Gamma_1 = (-\infty, 0]. \tag{6.8}$$

The densities have singularities at the branch points in the interior of their supports. The singularities are felt only at one side of the branch points. Consider first μ_0, whose density has a singularity at 0. However, the limiting value when 0 is approached from the positive real axis is finite. The change in behavior of μ_0 has to do with the fact that z_1 is analytic on $(0, 4)$ but not on $(-9/4, 0)$. Therefore we find by (1.12) that

$$\mathrm{d}\mu_0(\lambda) = \frac{1}{2\pi \mathrm{i}} \left(\frac{z_{1+}'(\lambda)}{z_{1+}(\lambda)} + \frac{z_{2+}'(\lambda)}{z_{2+}(\lambda)} - \frac{z_{1-}'(\lambda)}{z_{1-}(\lambda)} - \frac{z_{2-}'(\lambda)}{z_{2-}(\lambda)} \right) \mathrm{d}\lambda \tag{6.9}$$

on $(-9/4, 0)$ and

$$\mathrm{d}\mu_0(\lambda) = \frac{1}{2\pi \mathrm{i}} \left(\frac{z_{2+}'(\lambda)}{z_{2+}(\lambda)} - \frac{z_{2-}'(\lambda)}{z_{2-}(\lambda)} \right) \mathrm{d}\lambda \tag{6.10}$$

on $(0, 4)$.

For $\mu_{-1} = \mu_1$ a similar phenomenon happens at $\lambda = -9/4$. This is a consequence of the fact that z_1 has an analytic continuation into z_2 when we cross $(-\infty, -9/4)$, but it has an analytic continuation into z_4 when we cross $(-9/4, 0)$.

6.3. Example 3. As a final example, consider the symbol

$$a(z) = z^p + z^{-q} \tag{6.11}$$

with $p, q \geq 1$ and $\gcd(p, q) = 1$. This example appeared in [10], where the authors mentioned that Γ_0 is given by the star

$$\Gamma_0 = \{ r\omega^j \mid j = 1, \ldots, p+q,\ 0 \leq r \leq R \} \tag{6.12}$$

with $\omega = \mathrm{e}^{2\pi \mathrm{i}/(p+q)}$ and $R = (p+q)p^{-p/(p+q)}q^{-q/(p+q)}$. The other contours also have a star shape, namely

$$\Gamma_k = \{ (-1)^k r\omega^j \mid j = 1, \ldots, p+q,\ 0 \leq r < \infty \} \tag{6.13}$$

for $k \neq 0$. Note that the star Γ_k for $k \neq 0$ is unbounded.

In Figure 6.4 we plot the eigenvalues and the generalized eigenvalues for $p = 2$, $q = 3$, and $n = 50$. All the (generalized) eigenvalues appear to lie exactly on the

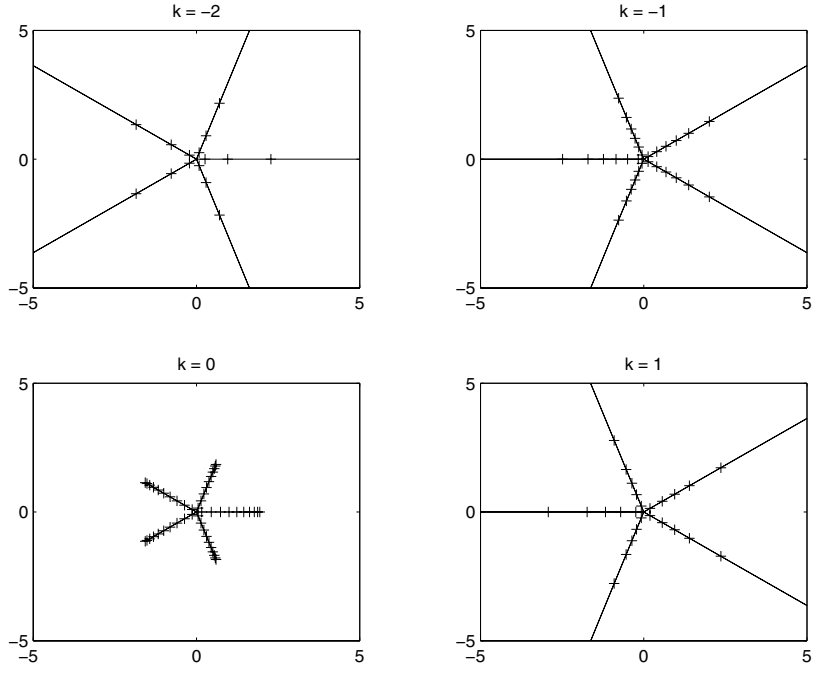


FIG. 6.4. *Illustration for Example* 3: *The contours* Γ_k *and the eigenvalues and generalized eigenvalues for* $T_{50}(a)$ *for the symbol* $a = z^2 + z^{-3}$.

contours. In the special case $p = 1$ it is known that the eigenvalues of $T_n(a)$ lie indeed precisely on the star (6.12) and are all simple (possibly except for 0) [4, Theorem 3.2]; see also [6] for a connection to Chebyshev-type quadrature.

6.4. Numerical stability. In Figures 6.2 and 6.4 the eigenvalues and the generalized eigenvalues of $T_{50}(a)$ were computed numerically. To control the stability of the numerical computation of the eigenvalues one needs to analyze the pseudospectrum. For banded Toeplitz matrices the pseudospectrum is well understood [12, Theorem 7.2]. To this date, a similar analysis of the pseudospectrum for the matrix pencil $(T_n(z^{-k}a), T_n(z^{-k}))$ has not been carried out. See [12, section X.45] for some remarks on the pseudospectrum for the generalized eigenvalue problem.

REFERENCES

[1] A. Böttcher and S. M. Grudsky, *Spectral Properties of Banded Toeplitz Matrices*, SIAM, Philadelphia, PA, 2005.

[2] A. Böttcher and S. M. Grudsky, *Can spectral values sets of Toeplitz band matrices jump?*, Linear Algebra Appl., 351–352 (2002), pp. 99–116.

[3] E. Coussement, J. Coussement, and W. Van Assche, *Asymptotic zero distribution for a class of multiple orthogonal polynomials*, Trans. Amer. Math. Soc., to appear.

[4] M. Eiermann and R. Varga, *Zeros and local extreme points of Faber polynomials associated with hypocycloidal domains*, Electron. Trans. Numer. Anal., 1 (1993), pp. 49–71.

[5] I. I. Hirschman, Jr., *The spectra of certain Toeplitz matrices*, Illinois J. Math., 11 (1967), pp. 145–159.

[6] A. Kuijlaars, *Chebyshev quadrature for measures with a strong singularity*, J. Comput. Appl. Math., 65 (1995), pp. 207–214.

[7] E. Nikishin and V. Sorokin, *Rational Approximations and Orthogonality*, Transl. Math. Monogr. 92, American Mathematical Society, Providence, RI, 1991.

[8] T. Ransford, *Potential Theory in the Complex Plane*, London Math. Soc. Stud. Texts 28, Cambridge University Press, Cambridge, UK, 1995.

[9] E. B. Saff and V. Totik, *Logarithmic Potentials with External Fields*, Grundlehren Math. Wiss. 316, Springer-Verlag, Berlin, 1997.

[10] P. Schmidt and F. Spitzer, *The Toeplitz matrices of an arbitrary Laurent polynomial*, Math. Scand., 8 (1960), pp. 15–38.

[11] P. Simeonov, *A weighted energy problem for a class of admissible weights*, Houston J. Math., 31 (2005), pp. 1245–1260.

[12] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra*, Princeton University Press, Princeton, NJ, 2005.

[13] J. L. Ullman, *A problem of Schmidt and Spitzer*, Bull. Amer. Math. Soc., 73 (1967), pp. 883–885.

[14] H. Widom, *On the eigenvalues of certain Hermitian operators*, Trans. Amer. Math. Soc., 88 (1958), pp. 491–522.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 197–222

ALGEBRAIC CHARACTERIZATIONS FOR POSITIVE REALNESS OF DESCRIPTOR SYSTEMS*

DELIN CHU† AND ROGER C. E. TAN†

Abstract. In this paper, algebraic characterizations for the positive realness of descriptor systems are studied. It is shown that the positive realness of descriptor systems can be determined by solving a linear matrix inequality, and hence the celebrated positive real lemma for standard state space systems is extended to descriptor systems. In addition, the lossless positive realness of both standard state space systems and descriptor systems is characterized explicitly based on the controllable staircase forms of standard state space systems and the generalized controllable staircase forms of descriptor systems, respectively.

Key words. positive realness, descriptor systems, linear matrix inequality

AMS subject classifications. 93B05, 93B40, 93B52, 65F35

DOI. 10.1137/060669061

1. Introduction.

In this paper we study the algebraic characterizations for the positive realness of descriptor systems in circuit and control theory. Throughout this paper, the following notation will be used:

- $M > 0$ (≥ 0) means that M is symmetric and positive definite (positive semidefinite);
- $M < 0$ (≤ 0) means $-M > 0$ (≥ 0);
- $j := \sqrt{-1}$, $\mathbf{C}_0 := \{s \mid s \in \mathbf{C},\ \mathrm{Re}(s) = 0\}$, $\mathbf{C}_+ := \{s \mid s \in \mathbf{C},\ \mathrm{Re}(s) > 0\}$.

Consider a system of the form

$$\begin{cases} E\dot{x}(t) = Ax(t) + Bu(t), \\ \quad y(t) = Cx(t) + Du(t), \end{cases} \tag{1}$$

where $E, A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, $C \in \mathbf{R}^{m\times n}$, and $D \in \mathbf{R}^{m\times m}$. System (1) is called a standard state space system if $E = I$. It is called a descriptor system (or a generalized state space system, or a singular system) if E is singular and the pencil (E, A) is regular (i.e., $\det(sE - A) \neq 0$ for some $s \in \mathbf{C}$).

DEFINITION 1. (i) [illegible] (1) [illegible] (E, A) [illegible]

$$\deg(sE - A) = \mathrm{rank}(E).$$

(ii) [illegible] (1) [illegible] (E, A) [illegible] ($\mathbf{C}_0 \cup \mathbf{C}_+$)

[illegible] 1. Let the pencil (E, A) be regular. There exist nonsingular matrices Q and P such that

$$QAP = \begin{bmatrix} A_{11} & 0 \\ 0 & I \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \end{matrix}, \quad QEP = \begin{bmatrix} I & 0 \\ 0 & \mathcal{N} \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \end{matrix}, \tag{2}$$

*Received by the editors September 5, 2006; accepted for publication (in revised form) by P. Benner May 22, 2007; published electronically February 27, 2008. This work was partially supported by NUS research grant R-146-000-087-112.

http://www.siam.org/journals/simax/30-1/66906.html

†Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543 (matchudl@nus.edu.sg, scitance@nus.edu.sg).

where $\mathcal{N}$ is nilpotent. Then system (1) is impulse-free if and only if $\mathcal{N} = 0$.

DEFINITION 2. [illegible] $E, A \in \mathbf{R}^{n\times n}$ [illegible] $B \in \mathbf{R}^{n\times m}$

(i) [illegible] $(E; A, B)$ [illegible]

$$\operatorname{rank}\begin{bmatrix} \alpha E - \beta A & B \end{bmatrix} = n \quad \forall (\alpha, \beta) \in \mathbf{C}^2 \backslash \{0, 0\}.$$

(ii) [illegible] $(E; A, B)$ [illegible]

$$\operatorname{rank}\begin{bmatrix} \alpha E - A & B \end{bmatrix} = n \quad \forall \alpha \in \mathbf{C}.$$

(iii) [illegible] (A, B) [illegible]

$$\operatorname{rank}\begin{bmatrix} \alpha I - A & B \end{bmatrix} = n \quad \forall \alpha \in \mathbf{C}.$$

Let

$$G(s) = D + C(sE - A)^{-1}B. \tag{3}$$

$G(s)$ is the transfer function of system (1). The positive realness of system (1) can be defined as follows.

DEFINITION 3 (see [3]).

(i) [illegible] (1) [illegible] $G(s)$ [illegible] $\mathbf{C}_+$ [illegible]

$$G(s) + (G(s))^H \geq 0 \quad \forall s \in \mathbf{C}_+.$$

(ii) [illegible] (1) [illegible]

$$G(j\omega) + (G(j\omega))^H = 0$$

[illegible] $\omega \in \mathbf{R}$ [illegible] $j\omega$ [illegible] $G(s)$

(iii) [illegible] (1) [illegible] $G(s)$ [illegible] $\mathbf{C}_0 \cup \mathbf{C}_+$ [illegible]

$$G(j\omega) + (G(j\omega))^H > 0 \quad \forall \omega \in \mathbf{R}.$$

(iv) [illegible] (1) [illegible] $G(s)$ [illegible]

$$G(j\infty) + (G(j\infty))^H > 0.$$

The concept of positive realness is motivated by network theory [7]. That is, a positive real rational function can be realized as the driving point impedance of a passive network, and, conversely, a passive network has a driving point impedance that is rational and positive real. As addressed in [5], reduced-order modeling techniques based on Krylov subspace iterations have become popular tools to tackle the large-scale dynamic systems that arise in the simulation of VLSI circuits [15]. These techniques are mostly applied to very large passive networks, and then it is important to check whether the reduced-order model preserves the passivity of the original network. Hence, the positive realness plays a major role in circuit theory.

The positive realness is also an important concept in control theory. It has many applications in model reference adaptive control, absolute stability of perturbed systems, robust control, inverse problem of optimal control, and flexible space structure; see [25, 35, 26, 27, 1, 36, 14, 28, 4, 6]. The list here is far from complete.

The positive realness of standard state space systems has been studied extensively; see [1, 3, 23, 38, 42, 2, 41]. Over the last three decades, several approaches for testing

the positive realness of a standard state space system have been developed. Among them, one powerful approach is to employ the celebrated positive real lemma, which characterizes the positive realness of a standard state space system by means of a linear matrix inequality (LMI).

The well-known positive real lemma can be stated as follows.

LEMMA 4 (positive real lemma [3]). [illegible] (1) [illegible] $E = I$ [illegible] (A, B) [illegible] (A^T, C^T) [illegible]

(i) [illegible] (1) [illegible] $X \in \mathbf{R}^{n\times n}$ [illegible]

$$\begin{bmatrix} A^TX + XA & XB - C^T \\ B^TX - C & -D - D^T \end{bmatrix} \le 0,\ X \ge 0. \tag{4}$$

(ii) [illegible] (1) [illegible] $X \in \mathbf{R}^{n\times n}$ [illegible]

$$\begin{bmatrix} A^TX + XA & XB - C^T \\ B^TX - C & -D - D^T \end{bmatrix} = 0,\ X \ge 0. \tag{5}$$

Lemma 4 was developed in the 1960s [25, 26, 27, 1, 36] and now is also known as the Kalman–Yakubovich lemma or the Yakubovich–Kalman–Popov–Anderson lemma. For a detailed analysis of the implication of Lemma 4, we refer the reader to [3, 36]. In addition, the analogy of Lemma 4 for system (1) with E nonsingular can be found in [18, 40].

The strict positive realness of standard state space systems is characterized algebraically in the following result, which is a generalization of Lemma 4(i).

LEMMA 5 (see [29, 34]). [illegible] (1) [illegible] $E = I$ [illegible] (A, B) [illegible] (A^T, C^T) [illegible] (1) [illegible] $X, L \in \mathbf{R}^{n\times n}$ [illegible] $W \in \mathbf{R}^{n\times m}$ [illegible]

$$\begin{bmatrix} A^TX + XA & XB - C^T \\ B^TX - C & -D - D^T \end{bmatrix} = -\begin{bmatrix} L^T \\ W^T \end{bmatrix}\begin{bmatrix} L & W \end{bmatrix},\ X \ge 0,$$

[illegible] (A^T, L^T) [illegible]

$$\operatorname{rank}\begin{bmatrix} A - j\omega I & B \\ L & W \end{bmatrix} = n + m \quad \forall \omega \in \mathbf{R}.$$

It is now well known that the descriptor systems have many important applications, for example, in circuit simulation, economic systems, network analysis, and biological systems. In fact, many systems in practical applications are singular in nature. The positive realness of descriptor systems has been considered recently under various assumptions. The following interesting results have been presented in [16, 32, 31, 45], respectively.

THEOREM 6 (see [16]). (a) [illegible] (1) [illegible] (E, A) [illegible] $\mathbf{C}_+$ [illegible]

$$\begin{bmatrix} A^TX + X^TA & X^TB - C^T \\ B^TX - C & -D - D^T \end{bmatrix} \le 0, \quad E^TX = X^TE \ge 0, \tag{6}$$

[illegible] X

(b) [illegible] (1) [illegible] (6) [illegible]

(i) [illegible] $(E;A,B)$ [illegible] $(E^T;A^T,C^T)$ [illegible]
(ii) [illegible] (1) [illegible]

$$A\ker(E) \subseteq \mathrm{Im}(E);$$

(iii) [illegible] $G(s)=\sum_{i=-\infty}^{p} s^i M_i$, [illegible] $G(s)$ [illegible] $s=\infty$ [illegible]

$$D + D^T \geq M_0 + M_0^T. \tag{7}$$

THEOREM 7 (see [32, 31]). [illegible] (1) [illegible] $X \in \mathbf{R}^{n\times n}$ [illegible] $Y \in \mathbf{R}^{n\times m}$ [illegible]

$$\begin{bmatrix} A^TX + X^TA & A^TY + X^TB - C^T \\ Y^TA + B^TX - C & Y^TB + B^TY - D - D^T \end{bmatrix} < 0,\ E^TX = X^TE \geq 0,\ E^TY = 0. \tag{8}$$

THEOREM 8 (see [45]). [illegible] (1) [illegible] $D + D^T > 0$ [illegible] $X \in \mathbf{R}^{n\times n}$ [illegible]

$$\begin{bmatrix} A^TX + X^TA & (C - B^TX)^T \\ C - B^TX & -(D + D^T) \end{bmatrix} < 0, \quad E^TX = X^TE \geq 0. \tag{9}$$

THEOREM 9 (see [45]). [illegible] (1) [illegible] $D + D^T > 0$ [illegible] $X \in \mathbf{R}^{n\times n}$ [illegible]

$$A^TX + X^TA + (C - B^TX)^T(D + D^T)^{-1}(C - B^TX) = 0, \quad E^TX = X^TE \geq 0. \tag{10}$$

The following statements are clear:

- Theorem 6 indicates that the solvability of the linear matrix inequality (6) is sufficient and, under the additional assumption (7), also necessary for the positive realness of descriptor system (1). However, in general, the assumption (7) may not be satisfied.
- Theorems 7–9 rely on the assumption that system (1) is impulse-free, which is strong. In fact, if system (1) is impulse-free, then there exist nonsingular matrices $P \in \mathbf{R}^{n\times n}$ and $Q \in \mathbf{R}^{n\times n}$ such that (2) holds with $\mathcal{N} = 0$. Consequently,

$$G(s) = (D - \mathcal{CB}) + C_1(sI - A_{11})^{-1}B_1, \tag{11}$$

where

$$QB = \begin{bmatrix} B_1 \\ \mathcal{B} \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \end{matrix}, \quad CP = \begin{matrix} n_1 & n_2 \\ \end{matrix} \begin{bmatrix} C_1 & \mathcal{C} \end{bmatrix}. \tag{12}$$

As a result, the impulse-free system (1) is extended strictly positive real if and only if the standard state space system

$$\begin{cases} \dot{x}(t) = A_{11}x(t) + B_1u(t), \\ y(t) = C_1x(t) + (D - \mathcal{CB})u(t) \end{cases}$$

is strictly positive real and $(D-\mathcal{CB}) + (D-\mathcal{CB})^T > 0$. Thus, existing results on the strictly positive realness of standard state space systems are applicable to the extended strictly positive realness of descriptor systems.

- If system (1) is minimal but not impulse-free, then, with the notation in (2) and (12), we have

$$G(s) = (D - \mathcal{C}\mathcal{B}) + C_1(sI - A_{11})^{-1}B_1 - \sum_{i=1}^{n_2-1} s^i \mathcal{C}\mathcal{N}^i\mathcal{B}$$
$$\neq (D - \mathcal{C}\mathcal{B}) + C_1(sI - A_{11})^{-1}B_1.$$

  Hence, the approaches used in [32, 31, 45] cannot be used to characterize all forms of the positive realness of system (1).

Based on the observations above, we conclude that in Theorems 6–9 there is obviously a gap between the sufficient condition and the necessary condition for the positive realness of descriptor systems (1). To the best of our knowledge, we are not aware of any conditions which are similar to (4) and are both necessary and sufficient for the positive realness of descriptor systems. In this sense, Lemma 4, which is the positive real lemma for standard state space systems, has not been extended to descriptor systems in the existing literature. In this paper we further develop the work in [16] and establish the positive real lemma for descriptor systems. It is surprising that our matrix inequality (14) in Theorem 13 of the next section is very close to the matrix inequality (8) in Theorem 7. However, these two inequalities play different roles in the sense that inequality (8) can only characterize the extended strictly positive realness of system (1) under the condition that system (1) is admissible, but cannot characterize the positive realness of system (1) in the general setting, while inequality (14) in the next section is not only necessary but also sufficient for the positive realness of system (1) in the general setting. In addition, the explicit algebraic characterizations for the lossless positive realness of both standard state space systems and descriptor systems will also be derived in this paper based on the controllable staircase forms of standard state space systems and the generalized controllable staircase forms of descriptor systems, respectively.

2. Main results. We derive the algebraic characterizations for the positive realness of descriptor systems in this section. For this purpose, we need some preliminary lemmas.

LEMMA 10 (see [3]). [illegible] $G(s)$ [illegible]

$$G(s) = G_1(s) + sM_1 + \sum_{i=2}^{p} s^i M_i, \quad G_1(\infty) < \infty.$$

[illegible] $G(s)$ [illegible] $G_1(s)$ [illegible] $M_1 \geq 0$ [illegible] $M_i = 0$, $i = 2, \ldots, p$

LEMMA 11. [illegible]

(13)
$$\mathcal{N} = \begin{array}{c} \\ \tau_1 \\ \tau_3 \\ \tau_2 \end{array} \begin{array}{c} \begin{array}{ccc} \tau_1 & \tau_2 & \tau_3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \end{array}, \quad \mathcal{B} = \begin{array}{c} \\ \tau_1 \\ \tau_3 \\ \tau_2 \end{array} \begin{bmatrix} \mathcal{B}_{12} \\ \mathcal{B}_{22} \\ \mathcal{B}_{32} \end{bmatrix}, \quad \mathcal{C} = \begin{array}{c} \begin{array}{ccc} \tau_1 & \tau_2 & \tau_3 \end{array} \\ \begin{bmatrix} \mathcal{C}_{21} & \mathcal{C}_{22} & \mathcal{C}_{23} \end{bmatrix} \end{array}, \ \tau_2 \leq \tau_3.$$

[illegible]

$$\operatorname{rank} \begin{bmatrix} \alpha\mathcal{N} - \beta I & \mathcal{B} \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \alpha\mathcal{N} - \beta I \\ \mathcal{C} \end{bmatrix} = \sum_{i=1}^{3} \tau_i \quad \forall \{\alpha, \beta\} \in \mathbf{C}^2 \backslash \{0, 0\}.$$

$$\mathcal{C}\mathcal{N}^i\mathcal{B} = 0, \quad i = 2, \ldots, \sum_{i=1}^{3} \tau_i - 1,$$

$\tau_2 = \tau_3$. Let $\tau = \sum_{i=1}^{3} \tau_i$. Note that the property

$$\operatorname{rank} \begin{bmatrix} \alpha\mathcal{N} - \beta I & \mathcal{B} \end{bmatrix} = \tau \quad \forall \{\alpha, \beta\} \in \mathbf{C}^2 \backslash \{0, 0\}$$

implies the controllability of $(\mathcal{N}, \mathcal{B})$. Consequently, we have from [43] that

$$\operatorname{rank} \begin{bmatrix} \mathcal{B} & \mathcal{N}\mathcal{B} & \cdots & \mathcal{N}^{\tau-1}\mathcal{B} \end{bmatrix} = \tau.$$

Because $\mathcal{N}$ is nilpotent and $\mathcal{N}^j = 0$ for all integer $j \geq \tau$, then

$$\begin{aligned}
&\mathcal{C}\mathcal{N}^i\mathcal{B} = 0, \quad i = 2, \ldots, \tau - 1 \\
\Longleftrightarrow\ &(\mathcal{C}\mathcal{N}^2)\mathcal{N}^i\mathcal{B} = 0, \quad i = 0, 1, \ldots, \tau - 1 \\
\Longleftrightarrow\ &\mathcal{C}\mathcal{N}^2 \begin{bmatrix} \mathcal{B} & \mathcal{N}\mathcal{B} & \cdots & \mathcal{N}^{\tau-1}\mathcal{B} \end{bmatrix} = 0 \\
\Longleftrightarrow\ &\mathcal{C}\mathcal{N}^2 = 0 \\
\Longleftrightarrow\ &\begin{bmatrix} \mathcal{C}_{22} & \mathcal{C}_{23} \end{bmatrix} \begin{bmatrix} I_{\tau_3 - \tau_2} \\ 0 \end{bmatrix} = 0 \quad (\text{since } \tau_2 \leq \tau_3).
\end{aligned}$$

However, it is given that $\operatorname{rank} \begin{bmatrix} \alpha\mathcal{N} - \beta I \\ \mathcal{C} \end{bmatrix} = \tau$ for all $\{\alpha, \beta\} \in \mathbf{C}^2 \backslash \{0, 0\}$; hence, $\mathcal{C}_{22}$ is of full column rank. Therefore, we obtain that $\mathcal{C}\mathcal{N}^i\mathcal{B} = 0$ $(i = 2, \ldots, \tau - 1)$ if and only if $\tau_2 = \tau_3$. □

LEMMA 12 (see [11, 10, 9]). $\mathcal{A} \in \mathbf{R}^{n \times n}$, $\mathcal{B} \in \mathbf{R}^{n \times m}$ $\mathcal{C} \in \mathbf{R}^{p \times n}$

(i)

$$\max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} \mathcal{A} - sI & \mathcal{B} \\ \mathcal{C} & 0 \end{bmatrix} = n$$

$$\mathcal{C}\mathcal{A}^i\mathcal{B} = 0, \quad i = 0, 1, 2, \ldots.$$

(ii) $(\mathcal{A}, \mathcal{B})$

$$\max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} \mathcal{A} - sI & \mathcal{B} \\ \mathcal{C} & 0 \end{bmatrix} = n$$

$$\mathcal{C} = 0.$$

We are now ready to characterize the positive realness of descriptor systems.

THEOREM 13. (1) $(E; A, B)$ $(E^T; A^T, C^T)$

(i) [illegible] (1) [illegible] $X \in \mathbf{R}^{n\times n}$ [illegible] $Y \in \mathbf{R}^{n\times m}$ [illegible]

$$\begin{cases} \begin{bmatrix} A^TX + X^TA & A^TY + X^TB - C^T \\ Y^TA + B^TX - C & B^TY + Y^TB - D - D^T \end{bmatrix} \le 0, \\ E^TX = X^TE \ge 0, \quad E^TY = 0. \end{cases} \tag{14}$$

(ii) [illegible] (1) [illegible] $X \in \mathbf{R}^{n\times n}$ [illegible] $Y \in \mathbf{R}^{n\times m}$ [illegible]

$$\begin{cases} \begin{bmatrix} A^TX + X^TA & A^TY + X^TB - C^T \\ Y^TA + B^TX - C & B^TY + Y^TB - D - D^T \end{bmatrix} = 0, \\ E^TX = X^TE \ge 0, \quad E^TY = 0. \end{cases} \tag{15}$$

(iii) [illegible] (1) [illegible] $X, L \in \mathbf{R}^{n\times n}$, $Y \in \mathbf{R}^{n\times m}$ [illegible] $W \in \mathbf{R}^{n\times m}$ [illegible]

$$\begin{cases} \begin{bmatrix} A^TX + X^TA & A^TY + X^TB - C^T \\ Y^TA + B^TX - C & B^TY + Y^TB - D - D^T \end{bmatrix} = -\begin{bmatrix} L^T \\ W^T \end{bmatrix} \begin{bmatrix} L & W \end{bmatrix}, \\ E^TX = X^TE \ge 0, \quad E^TY = 0. \end{cases} \tag{16}$$

$(E^T; A^T, L^T)$ [illegible]

$$\operatorname{rank} \begin{bmatrix} A - j\omega E & B \\ L & W \end{bmatrix} = n + m \quad \forall \omega \in \mathbf{R}. \tag{17}$$

[illegible]. We prove the necessity first and then the sufficiency.

By matrix pencil theory [17], there exist nonsingular matrices $P, Q \in \mathbf{R}^{n\times n}$ such that (2), (12), and (13) hold. Consequently,

$$\begin{aligned} G(s) &= D + C_1(sI - A_{11})^{-1}B_1 + \mathcal{C}(s\mathcal{N} - I)^{-1}\mathcal{B} \\ &= D - \mathcal{C}\mathcal{B} + C_1(sI - A_{11})^{-1}B_1 - \sum_{i=1}^{n_2-1} s^i \mathcal{C}\mathcal{N}^i\mathcal{B}. \end{aligned}$$

[illegible] (i). Let $G(s)$ be positive real. By Lemma 10, $D - \mathcal{C}\mathcal{B} + C_1(sI - A_{11})^{-1}B_1$ is positive real, and

$$\mathcal{C}\mathcal{N}\mathcal{B} \le 0, \quad \mathcal{C}\mathcal{N}^i\mathcal{B} = 0, \ i = 2, \ldots, n_2 - 1.$$

Since $(E; A, B)$ and $(E^T; A^T, C^T)$ are controllable, it is easy to verify that (A_{11}, B_1) and (A_{11}^T, C_1^T) are controllable, and

$$\operatorname{rank} \begin{bmatrix} \alpha\mathcal{N} - \beta I & \mathcal{B} \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \alpha\mathcal{N} - \beta I \\ \mathcal{C} \end{bmatrix} = n_2 = \sum_{i=1}^{3} \tau_i \quad \forall \{\alpha, \beta\} \in \mathbf{C}^2 \backslash \{0, 0\}. \tag{18}$$

Then by Lemmas 4(i) and 11, there exist $X_{11}, L_1 \in \mathbf{R}^{n_1\times n_1}$ and $W_1 \in \mathbf{R}^{n_1\times m}$ such that

$$\begin{cases} A_{11}^T X_{11} + X_{11}^T A_{11} = -L_1^T L_1, \\ X_{11}^T B_1 - C_1^T = -L_1^T W_1, \\ -(D - \mathcal{CB}) - (D - \mathcal{CB})^T = -W_1^T W_1, \\ X_{11} \geq 0, \end{cases} \tag{19}$$

and $\tau_2 = \tau_3$.

Because (18) means $\operatorname{rank}(\mathcal{C}_{22}) = \operatorname{rank}(\mathcal{B}_{32}) = \tau_2$, and the property $\tau_2 = \tau_3$ implies that $\mathcal{CNB} \leq 0$ is equivalent to

$$\mathcal{C}_{22}\mathcal{B}_{32} = \mathcal{B}_{32}^T\mathcal{C}_{22}^T \leq 0,$$

thus

$$\mathcal{Z}_{23} := -\mathcal{C}_{22}^T\mathcal{B}_{32}^T(\mathcal{B}_{32}\mathcal{B}_{32}^T)^{-1}, \quad \mathcal{X} := \begin{matrix} \\ \\ \\ \end{matrix} \begin{matrix} \tau_1 & \tau_2 & \tau_2 \\ \end{matrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \mathcal{Z}_{23} \\ 0 & -\mathcal{Z}_{23}^T & 0 \end{bmatrix} \begin{matrix} \}\tau_1 \\ \}\tau_2 \\ \}\tau_3 \end{matrix}$$

satisfy

$$\mathcal{Z}_{23} = \mathcal{Z}_{23}^T \geq 0$$

and

$$\begin{cases} \mathcal{N}^T\mathcal{X} = \mathcal{X}^T\mathcal{N} \geq 0, \\ \mathcal{X} + \mathcal{X}^T = 0, \\ \mathcal{N}^T(\mathcal{C}^T - \mathcal{X}^T\mathcal{B}) = 0. \end{cases} \tag{20}$$

Moreover, we have

(21)

$$\begin{cases} \begin{bmatrix} A_{11} & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} X_{11} & 0 \\ 0 & \mathcal{X} \end{bmatrix} + \begin{bmatrix} X_{11} & 0 \\ 0 & \mathcal{X} \end{bmatrix}^T \begin{bmatrix} A_{11} & 0 \\ 0 & I \end{bmatrix} = -\begin{bmatrix} L_1^T \\ 0 \end{bmatrix} \begin{bmatrix} L_1 & 0 \end{bmatrix}, \\ \begin{bmatrix} A_{11} & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} 0 \\ \mathcal{C}^T - \mathcal{X}^T\mathcal{B} \end{bmatrix} + \begin{bmatrix} X_{11} & 0 \\ 0 & \mathcal{X} \end{bmatrix}^T \begin{bmatrix} B_1 \\ \mathcal{B} \end{bmatrix} - \begin{bmatrix} C_1 & \mathcal{C} \end{bmatrix}^T \\ \qquad = -\begin{bmatrix} L_1^T \\ 0 \end{bmatrix} W_1, \\ \begin{bmatrix} B_1 \\ \mathcal{B} \end{bmatrix}^T \begin{bmatrix} 0 \\ \mathcal{C}^T - \mathcal{X}^T\mathcal{B} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathcal{C}^T - \mathcal{X}^T\mathcal{B} \end{bmatrix}^T \begin{bmatrix} B_1 \\ \mathcal{B} \end{bmatrix} - D - D^T = -W_1^T W_1, \\ \begin{bmatrix} I & 0 \\ 0 & \mathcal{N} \end{bmatrix}^T \begin{bmatrix} X_{11} & 0 \\ 0 & \mathcal{X} \end{bmatrix} = \begin{bmatrix} X_{11} & 0 \\ 0 & \mathcal{X} \end{bmatrix}^T \begin{bmatrix} I & 0 \\ 0 & \mathcal{N} \end{bmatrix} \geq 0, \\ \begin{bmatrix} I & 0 \\ 0 & \mathcal{N} \end{bmatrix}^T \begin{bmatrix} 0 \\ \mathcal{C}^T - \mathcal{X}^T\mathcal{B} \end{bmatrix} = 0. \end{cases}$$

Define

$$X = Q^T \begin{bmatrix} X_{11} & 0 \\ 0 & \mathcal{X} \end{bmatrix} P^{-1}, \quad Y = Q^T \begin{bmatrix} 0 \\ \mathcal{C}^T - \mathcal{X}^T\mathcal{B} \end{bmatrix} \tag{22}$$

and

$$L = \begin{bmatrix} L_1 & 0 \\ 0 & 0 \end{bmatrix} P^{-1} \in \mathbf{R}^{n\times n}, \quad W = \begin{bmatrix} W_1 \\ 0 \end{bmatrix} \in \mathbf{R}^{n\times m}. \tag{23}$$

Then we have by using (2), (12), (13), and (21) that

$$\begin{cases} A^TX + X^TA = -L^TL, \\ A^TY + X^TB - C^T = -L^TW, \\ B^TY + Y^TB - D - D^T = -W^TW, \\ E^TX = X^TE \geq 0, \\ E^TY = 0, \end{cases}$$

i.e., (14) holds.

(ii). We have from the necessity proof of part (i) that

$$G(s) = D - \mathcal{CB} + C_1(sI - A_{11})^{-1}B_1 - s\mathcal{CNB}, \quad \tau_2 = \tau_3, \quad \mathcal{CNB} \leq 0,$$

and $D - \mathcal{CB} + C_1(sI - A_{11})^{-1}B_1$ is positive real. Note that

$$D-\mathcal{CB}+C_1(j\omega I-A_{11})^{-1}B_1+(D-\mathcal{CB}+C_1(j\omega I-A_{11})^{-1}B_1)^H = G(j\omega)+(G(j\omega))^H$$

for all $\omega \in \mathbf{R}$ with $j\omega$ not a pole of $G(s)$, and $D - \mathcal{CB} + C_1(sI - A_{11})^{-1}B_1$ and $G(s)$ have the same poles; we get that $D - \mathcal{CB} + C_1(sI - A_{11})^{-1}B_1$ is lossless positive real. Thus, by Lemma 4(ii), there exists $X_{11} \in \mathbf{R}^{n_1\times n_1}$ such that

$$\begin{cases} A_{11}^TX_{11} + X_{11}^TA_{11} = 0, \\ X_{11}^TB_1 - C_1^T = 0, \\ -(D - \mathcal{CB}) - (D - \mathcal{CB})^T = 0, \\ X_{11} \geq 0. \end{cases}$$

Let X and Y be defined by (22). Then it is easy to verify that

$$\begin{cases} A^TX + X^TA = 0, \\ A^TY + X^TB - C^T = 0, \\ B^TY + Y^TB - D - D^T = 0, \\ E^TX = X^TE \geq 0, \\ E^TY = 0, \end{cases}$$

i.e., (15) holds.

(iii). We know from the necessity proof of part (i) that $\tau_2 = \tau_3$, $\mathcal{CNB} \leq 0$, and $D-\mathcal{CB}+C_1(sI-A_{11})^{-1}B_1$ is positive real. Moreover, $D-\mathcal{CB}+C_1(sI-A_{11})^{-1}B_1$ is strictly positive real. So, according to Lemma 5, there exist $X_{11}, L_1 \in \mathbf{R}^{n_1\times n_1}$ and $W_1 \in \mathbf{R}^{n_1\times m}$ such that (19) is satisfied, (A_{11}^T, L_1^T) is controllable, and

$$\operatorname{rank}\begin{bmatrix} A_{11} - j\omega I & B_1 \\ L_1 & W_1 \end{bmatrix} = n_1 + m \quad \forall \omega \in \mathbf{R}.$$

Let X, Y, L, and W be defined by (22) and (23); we have that (16) holds,

$$\begin{aligned}\operatorname{rank}\begin{bmatrix} \alpha E^T - A^T & L^T \end{bmatrix} &= \operatorname{rank}\begin{bmatrix} \alpha I - A_{11}^T & 0 & L_1^T \\ 0 & \alpha\mathcal{N}^T - I & 0 \end{bmatrix} \\ &= \operatorname{rank}\begin{bmatrix} \alpha I - A_{11}^T & L_1^T \end{bmatrix} + \operatorname{rank}(\alpha\mathcal{N}^T - I) \\ &= n_1 + n_2 = n \quad \forall \alpha \in \mathbf{C},\end{aligned}$$

equivalently, $(E^T; A^T, L^T)$ is R-controllable, and moreover

$$\begin{aligned}\operatorname{rank}\begin{bmatrix} A - j\omega E & B \\ L & W \end{bmatrix} &= \operatorname{rank}\begin{bmatrix} A_{11} - j\omega I & 0 & B_1 \\ 0 & I - j\omega\mathcal{N} & \mathcal{B} \\ L_1 & 0 & W_1 \end{bmatrix} \\ &= \operatorname{rank}\begin{bmatrix} A_{11} - j\omega I & B_1 \\ L_1 & W_1 \end{bmatrix} + n_2 \\ &= (n_1 + m) + n_2 = n + m \quad \forall \omega \in \mathbf{R}.\end{aligned}$$

[illegible] (i). In this case, (14) implies that

$$\begin{bmatrix} A^TX + X^TA & A^TY + X^TB - C^T \\ Y^TA + B^TX - C & B^TY + Y^TB - D - D^T \end{bmatrix} = -\begin{bmatrix} L^T \\ W^T \end{bmatrix}\begin{bmatrix} L & W \end{bmatrix} \tag{24}$$

for some $L \in \mathbf{R}^{n\times n}$ and $W \in \mathbf{R}^{n\times m}$.

Let us first show that $G(s)$ is analytic in $\mathbf{C}_+$. For this purpose, we show that the pencil (E, A) has no finite generalized eigenvalues in $\mathbf{C}_+$.

By contradiction, assume that $\lambda \in \mathbf{C}_+$ is a finite generalized eigenvalue of the pencil (E, A) and $y \in \mathbf{C}^n$ is a corresponding eigenvector. Then

$$Ay = \lambda Ey, \quad y \neq 0. \tag{25}$$

So, (24) and $E^TX = X^TE \geq 0$ yield that

$$\begin{aligned}0 &\geq -y^HL^TLy \\ &= y^H(A^TX + X^TA)y \\ &= \bar{\lambda}y^HE^TXy + \lambda y^HX^TEy \\ &= 2\mathrm{Re}(\lambda)y^HE^TXy \\ &\geq 0, \qquad (26)\end{aligned}$$

which together with $\lambda \in \mathbf{C}_+$ gives that

$$X^TEy = E^TXy = 0. \tag{27}$$

In return, we get by using (25), (26), and (27) that

$$X^TAy = 0, \quad Ly = 0 \tag{28}$$

and

$$A^T Xy = (A^T X + X^T A)y - X^T Ay = -L^T Ly - X^T Ay = 0. \tag{29}$$

Note that the pencil (E, A) is regular; thus $\operatorname{rank}\left[\begin{smallmatrix} E^T \\ A^T \end{smallmatrix}\right] = n$. In addition, we have shown that $\left[\begin{smallmatrix} E^T \\ A^T \end{smallmatrix}\right] Xy = 0$. Hence, we get $Xy = 0$. Consequently,

$$Cy = (Y^T A + B^T X + W^T L)y = Y^T Ay = \lambda Y^T Ey = \lambda (E^T Y)^T y = 0.$$

This and (25) give

$$\begin{bmatrix} A - \lambda E \\ C \end{bmatrix} y = 0.$$

Because the controllability of $(E^T; A^T, C^T)$ implies $\operatorname{rank}\left[\begin{smallmatrix} A - \lambda E \\ C \end{smallmatrix}\right] = n$, we obtain

$$y = 0,$$

which contradicts the fact $y \neq 0$. Therefore, the pencil (E, A) has no finite generalized eigenvalues in $\mathbf{C}_+$ and thus $G(s)$ is analytic in $\mathbf{C}_+$.

Next, we have using $E^T X = X^T E$ that

$$(sE - A)^H X + X^T (sE - A) = L^T L + 2\mathrm{Re}(s) E^T X$$

and

$$\begin{aligned} G(s) &= D + C(sE - A)^{-1} B = D + (Y^T A + B^T X + W^T L)(sE - A)^{-1} B \\ &= D + Y^T A(sE - A)^{-1} B + W^T L(sE - A)^{-1} B \\ &\quad + ((sE - A)^{-1} B)^H (sE - A)^H X (sE - A)^{-1} B \quad \forall s \in \mathbf{C}_+, \end{aligned}$$

which together with $E^T X = X^T E$ and $E^T Y = 0$ yield that

$$\begin{aligned} G(s) + (G(s))^H &= B^T Y + Y^T B + Y^T A(sE - A)^{-1} B + ((sE - A)^{-1} B)^H A^T Y \\ &\quad + [W + L(sE - A)^{-1} B]^H [W + L(sE - A)^{-1} B] \\ &\quad + 2\mathrm{Re}(s)((sE - A)^{-1} B)^H E^T X (sE - A)^{-1} B \\ &= sY^T E(sE - A)^{-1} B + (sY^T E(sE - A)^{-1} B)^H \\ &\quad + [W + L(sE - A)^{-1} B]^H [W + L(sE - A)^{-1} B] \\ &\quad + 2\mathrm{Re}(s)((sE - A)^{-1} B)^H E^T X (sE - A)^{-1} B \\ &= [W + L(sE - A)^{-1} B]^H [W + L(sE - A)^{-1} B] \\ &\quad + 2\mathrm{Re}(s)((sE - A)^{-1} B)^H E^T X (sE - A)^{-1} B \end{aligned} \tag{30}$$

for any $s \in \mathbf{C}_+$. Hence, $G(s) + (G(s))^H \geq 0$ because $E^T X = X^T E \geq 0$. Therefore, $G(s)$ is positive real.

(ii). By part (i) we know that $G(s)$ is positive real. Moreover, same as the derivation of (30), we have

$$G(j\omega) + (G(j\omega))^H = 2\mathrm{Re}(j\omega)((j\omega E - A)^{-1}B)^H E^T X (j\omega E - A)^{-1}B = 0$$

for all $\omega \in \mathbf{R}$ with $j\omega$ not a pole of $G(s)$. So, $G(s)$ is lossless positive real.

(iii). By part (i), $G(s)$ is positive real. Let $\lambda \in \mathbf{C}_0$ be a finite generalized eigenvalue of the pencil (E, A) and let $y \in \mathbf{C}^n$ be a corresponding eigenvector. Then (26) holds and thus $Ly = 0$. So, we obtain

$$\begin{bmatrix} \lambda E - A \\ L \end{bmatrix} y = 0,$$

which together with the R-controllability of $(E^T; A^T, L^T)$ yields that

$$y = 0.$$

This contradicts the fact $y \neq 0$. Therefore, the pencil (E, A) has no finite generalized eigenvalues in $\mathbf{C}_0$. Hence, $G(s)$ is analytic in $\mathbf{C}_+ \cup \mathbf{C}_0$ because $G(s)$ is positive real.

Next, we also have by using (17) and (30)

$$\begin{aligned} G(j\omega) + (G(j\omega))^H &= [W + L(j\omega E - A)^{-1}B]^H [W + L(j\omega E - A)^{-1}B] \\ &> 0 \quad \forall \omega \in \mathbf{R}. \end{aligned}$$

Hence, $G(s)$ is strictly positive real. □

If $E = I$, then Theorem 13(i) and (ii) reduce to Lemma 4(i) and (ii), respectively. Hence, Theorem 13(i) and (ii) can be regarded as the positive real lemma for descriptor systems. Therefore, we have extended the well-known positive real lemma from standard state space systems to descriptor systems and show that the positive realness of descriptor systems can be tested by solving a linear matrix inequality of the form (14).

The following two corollaries are trivial consequences of the sufficiency proofs in Theorem 13.

COROLLARY 14. (1) (E, A)

(i) $X \in \mathbf{R}^{n\times n}$ $Y \in \mathbf{R}^{n\times m}$ (14) (1)

(ii) $X \in \mathbf{R}^{n\times n}$ $Y \in \mathbf{R}^{n\times m}$ (15) (1)

(iii) $X, L \in \mathbf{R}^{n\times n}$ $Y \in \mathbf{R}^{n\times m}$ $W \in \mathbf{R}^{m\times n}$ (16) (17) $(E^T; A^T, L)$ (1)

COROLLARY 15. (1) $(E^T; A^T, C^T)$

$$\mathrm{rank}\begin{bmatrix} \alpha E - A \\ C \end{bmatrix} = n \quad \forall \alpha \in \mathbf{C}_0 \cup \mathbf{C}_+,$$

$$\mathrm{rank}\begin{bmatrix} \alpha E - A \\ C \end{bmatrix} = n$$

$\alpha \in \mathbf{C}$ (E, A)

(i) $X \in \mathbf{R}^{n\times n}$ $Y \in \mathbf{R}^{n\times m}$ (14) (1).

(ii) $X \in \mathbf{R}^{n\times n}$ $Y \in \mathbf{R}^{n\times m}$ (15) (1).

(iii) $X, L \in \mathbf{R}^{n\times n}$, $Y \in \mathbf{R}^{n\times m}$, $W \in \mathbf{R}^{m\times n}$ (16) (17) $(E^T; A^T, L)$ (1).

In the following we derive explicit algebraic characterizations for the lossless positive realness of both standard state space systems and descriptor systems based on the controllable staircase form theory.

THEOREM 16. (1) $E = I$ (A, B) (A^T, C^T) $P \in \mathbf{R}^{2n\times 2n}$ $(P^T\left[\begin{smallmatrix} A & 0 \\ 0 & -A^T \end{smallmatrix}\right]P, P^T\left[\begin{smallmatrix} B \\ C^T \end{smallmatrix}\right])$ [39]:

$$P^T \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} P = \begin{matrix} \\ \mu_1 \\ \mu_2 \end{matrix} \begin{matrix} \begin{matrix} \mu_1 & \mu_2 \end{matrix} \\ \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix} \end{matrix}, \quad P^T \begin{bmatrix} B \\ C^T \end{bmatrix} = \begin{matrix} \\ \mu_1 \\ \mu_2 \end{matrix} \begin{matrix} \\ \begin{bmatrix} \Psi_1 \\ 0 \end{bmatrix} \end{matrix}, \tag{31}$$

(Φ_{11}, Ψ_1)

$$\begin{bmatrix} C & -B^T \end{bmatrix} P = \begin{matrix} \begin{matrix} \mu_1 & \mu_2 \end{matrix} \\ \begin{bmatrix} \mathcal{K}_1 & \mathcal{K}_2 \end{bmatrix} \end{matrix}, \quad P = \begin{matrix} \begin{matrix} \mu_1 & \mu_2 \end{matrix} \\ \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \end{matrix} \begin{matrix} \\ \}n \\ \}n \end{matrix}. \tag{32}$$

(1)

$$D + D^T = 0, \quad \mathcal{K}_1 = 0, \quad P_{11}^T P_{21} \geq 0. \tag{33}$$

$\mu_1 = \mu_2 = n$, P_{11} X (5)

$$X = P_{21} P_{11}^{-1}. \tag{34}$$

. We prove the necessity first and then the sufficiency.

. Since system (1) is lossless positive real, by Lemma 4(ii),

$$D + D^T = 0,$$

and there exists a matrix $X \in \mathbf{R}^{n\times n}$ such that

$$A^T X + XA = 0, \quad C = B^T X, \quad X \geq 0.$$

Consequently, we obtain

$$CA^i B = B^T (-A^T)^i C^T, \quad i = 0, 1, 2, \ldots,$$

i.e.,

$$\begin{bmatrix} C & -B^T \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix}^i \begin{bmatrix} B \\ C^T \end{bmatrix} = 0, \quad i = 0, 1, 2, \ldots.$$

So, we have using Lemma 12(i) that

$$\max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sI-A & 0 & B \\ 0 & sI+A^T & C^T \\ C & -B^T & 0 \end{bmatrix} = 2n,$$

which is equivalent to

$$\mu_1+\mu_2 = 2n = \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sI-\Phi_{11} & -\Phi_{12} & \Psi_1 \\ 0 & sI-\Phi_{22} & 0 \\ \mathcal{K}_1 & \mathcal{K}_2 & 0 \end{bmatrix}$$

$$= \mu_2 + \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sI-\Phi_{11} & \Psi_1 \\ \mathcal{K}_1 & 0 \end{bmatrix},$$

i.e.,

$$\max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sI-\Phi_{11} & \Psi_1 \\ \mathcal{K}_1 & 0 \end{bmatrix} = \mu_1.$$

Note that (Φ_{11}, Ψ_1) is controllable, and thus Lemma 12(ii) gives

$$\mathcal{K}_1 = 0.$$

In addition,

$$\max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sI-A & 0 & B \\ 0 & sI+A^T & C^T \\ X^T & -I & 0 \end{bmatrix}$$

$$= \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sI-A & 0 & B \\ -X^T(sI-A)+(sI+A^T)X^T & sI+A^T & C^T-X^TB \\ 0 & -I & 0 \end{bmatrix}$$

$$= \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sI-A & 0 & B \\ 0 & sI+A^T & 0 \\ 0 & -I & 0 \end{bmatrix} = 2n,$$

which gives that

$$\mu_1+\mu_2 = 2n = \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sI-\Phi_{11} & -\Phi_{12} & \Psi_1 \\ 0 & sI-\Phi_{22} & 0 \\ X^TP_{11}-P_{21} & X^TP_{12}-P_{22} & 0 \end{bmatrix}$$

$$= \mu_2 + \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sI-\Phi_{11} & \Psi_1 \\ X^TP_{11}-P_{21} & 0 \end{bmatrix},$$

i.e.,

$$\max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sI-\Phi_{11} & \Psi_1 \\ X^TP_{11}-P_{21} & 0 \end{bmatrix} = \mu_1,$$

which together with the controllability of (Φ_{11}, Ψ_1) yields that

$$X^T P_{11} - P_{21} = 0.$$

Therefore, the property $X^T = X \geq 0$ leads to

$$P_{11}^T P_{21} = P_{11}^T X^T P_{11} \geq 0.$$

. We have using the property $\mathcal{K}_1 = 0$ that

$$\max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} sI - A & 0 & B \\ 0 & sI + A^T & C^T \\ C & -B^T & 0 \end{bmatrix}$$

$$= \max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} sI - \Phi_{11} & -\Phi_{12} & \Psi_1 \\ 0 & sI - \Phi_{22} & 0 \\ 0 & \mathcal{K}_2 & 0 \end{bmatrix} = \mu_1 + \mu_2 = 2n.$$

So, it follows from Lemma 12(i) that

$$\begin{bmatrix} C & -B^T \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix}^i \begin{bmatrix} B \\ C^T \end{bmatrix} = 0, \quad i = 0, 1, 2, \ldots.$$

Thus,

$$CA^i \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$$

$$= B^T(-A^T)^i \begin{bmatrix} C^T & (-A^T)C^T & \cdots & (-A^T)^{n-1}C^T \end{bmatrix}, \quad i = 0, 1, 2, \ldots,$$

which yields that for $i = 0, 1, 2, \ldots,$

$$CA^i \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} \begin{bmatrix} B^T \\ B^T A^T \\ \vdots \\ B^T (A^T)^{n-1} \end{bmatrix}$$

$$= B^T(-A^T)^i \begin{bmatrix} C^T & (-A^T)C^T & \cdots & (-A^T)^{n-1}C^T \end{bmatrix} \begin{bmatrix} B^T \\ B^T A^T \\ \vdots \\ B^T (A^T)^{n-1} \end{bmatrix}.$$

Since (A, B) is controllable, it is well known [43] that

$$\operatorname{rank} \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} = n,$$

so,

$$\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} \begin{bmatrix} B^T \\ B^T A^T \\ \vdots \\ B^T (A^T)^{n-1} \end{bmatrix}$$

is nonsingular and, further,

(35)
$$CA^i = B^T(-A^T)^i X,$$
$$\text{i.e., } \begin{bmatrix} X^T & -I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix}^i \begin{bmatrix} B \\ C^T \end{bmatrix} = X^T A^i B - (-A^T)^i C^T = 0, \quad i = 0, 1, 2, \ldots,$$

in particular,

$$C = B^T X, \tag{36}$$

where

$$X = \begin{bmatrix} C^T & (-A^T)C^T & \cdots & (-A^T)^{n-1}C^T \end{bmatrix} \begin{bmatrix} B^T \\ B^T A^T \\ \vdots \\ B^T (A^T)^{n-1} \end{bmatrix} \left(\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} \begin{bmatrix} B^T \\ B^T A^T \\ \vdots \\ B^T (A^T)^{n-1} \end{bmatrix} \right)^{-1}.$$

Hence, Lemma 12(i) implies that

$$\max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} sI - A & 0 & B \\ 0 & sI + A^T & C^T \\ X^T & -I & 0 \end{bmatrix} = 2n. \tag{37}$$

In return, we obtain that

$$\begin{aligned} &\max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} sI - \Phi_{11} & \Psi_1 \\ X^T P_{11} - P_{21} & 0 \end{bmatrix} \\ &= \max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} sI - \Phi_{11} & -\Phi_{12} & \Psi_1 \\ 0 & sI - \Phi_{22} & 0 \\ X^T P_{11} - P_{21} & X^T P_{12} - P_{22} & 0 \end{bmatrix} - \mu_2 \\ &= \max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} sI - A & 0 & B \\ 0 & sI + A^T & C^T \\ X^T & -I & 0 \end{bmatrix} - \mu_2 \\ &= 2n - \mu_2 = (\mu_1 + \mu_2) - \mu_2 \\ &= \mu_1, \end{aligned}$$

which together with Lemma 12(ii) and the controllability of (Φ_{11}, Ψ_1) implies that

$$X^T P_{11} - P_{21} = 0 \tag{38}$$

and

$$\begin{aligned}
\left[\begin{array}{cc} X^T & -I \end{array}\right] P &= \left[\begin{array}{cc} 0 & X^T P_{12} - P_{22} \end{array}\right], \\
\left[\begin{array}{cc} X^T & -I \end{array}\right] &= \left[\begin{array}{cc} 0 & X^T P_{12} - P_{22} \end{array}\right] P^T \\
&= \left[\begin{array}{cc} (X^T P_{12} - P_{22}) P_{12}^T & (X^T P_{12} - P_{22}) P_{22}^T \end{array}\right].
\end{aligned}$$

So, we have by taking $P_{22} \in \mathbf{R}^{n \times \mu_2}$ into account that

$$\begin{aligned}
n &= (\operatorname{rank} \left[\begin{array}{cc} X^T & -I \end{array}\right] P) = \operatorname{rank}(X^T P_{12} - P_{22}) \leq \mu_2 = 2n - \mu_1, \\
-I &= (X^T P_{12} - P_{22}) P_{22}^T, \\
n &= \operatorname{rank}(-I) = \operatorname{rank}((X^T P_{12} - P_{22}) P_{22}^T) \leq \operatorname{rank}(P_{22}^T) \leq n,
\end{aligned}$$

i.e.,

$$n \leq \mu_2 = 2n - \mu_1, \quad n = \operatorname{rank}(P_{22}). \tag{39}$$

Because (A, B) is controllable, μ_1 in the controllable staircase form (31) must satisfy

$$n \leq \mu_1, \tag{40}$$

and we get using (39) and (40) that

$$n = \mu_1 = \mu_2, \quad \operatorname{rank}(P_{22}) = n = \mu_2.$$

This means that P_{22} is nonsingular and so P_{11} is also nonsingular as P is orthogonal. As a result, (38) and $P_{11}^T P_{21} \geq 0$ lead to

$$X^T = P_{21} P_{11}^{-1} = P_{11}^{-T} (P_{11}^T P_{21}) P_{11}^{-1} \geq 0, \text{ i.e., } X = P_{21} P_{11}^{-1} \geq 0. \tag{41}$$

Furthermore, we also have using $C^T = X^T B = XB$ that

$$\begin{aligned}
2n &= \max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} sI - A & 0 & B \\ 0 & sI + A^T & C^T \\ X^T & -I & 0 \end{bmatrix} = \max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} sI - A & 0 & B \\ 0 & sI + A^T & C^T \\ X & -I & 0 \end{bmatrix} \\
&= \max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} sI - A & 0 & B \\ (sI + A^T)X - X(sI - A) & sI + A^T & 0 \\ 0 & -I & 0 \end{bmatrix} \\
&= n + \max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} sI - A & B \\ A^T X + XA & 0 \end{bmatrix},
\end{aligned}$$

i.e.,

$$n = \max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} sI - A & B \\ A^T X + XA & 0 \end{bmatrix}.$$

However, (A, B) is controllable, so we must have by Lemma 12(ii) that

$$A^T X + XA = 0. \tag{42}$$

Therefore, the lossless positive realness of system (1) follows directly from the property $D + D^T = 0$, equalities (36), (41), and (42), and Lemma 4(ii). □

We turn to consider the explicit verification of the lossless positive realness of descriptor systems.

THEOREM 17. [illegible] (1) [illegible] $(E; A, B)$ [illegible] $(E^T; A^T, C^T)$ [illegible]

- [illegible] [13, 39], $sE - A$ [illegible] $sE^T - A^T$

$$\begin{cases} U(sE - A)V = \begin{matrix} \\ \end{matrix} \begin{matrix} n_1 & n_2 \end{matrix} \\ \qquad \begin{bmatrix} sE_{11} - A_{11} & sE_{12} - A_{12} \\ 0 & sE_{22} - A_{22} \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \end{matrix}, \\ \mathcal{V}^T V^T (sE^T - A^T) U^T \mathcal{U}^T = \begin{matrix} \\ \end{matrix} \begin{matrix} n_1 & n_2 \end{matrix} \\ \qquad \begin{bmatrix} s\mathcal{E}_{11} - \mathcal{A}_{11} & s\mathcal{E}_{12} - \mathcal{A}_{12} \\ 0 & s\mathcal{E}_{22} - \mathcal{A}_{22} \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \end{matrix}, \end{cases} \tag{43}$$

[illegible] $U, \mathcal{U}, V, \mathcal{V}$ [illegible] E_{11} [illegible] $\mathcal{E}_{11}$ [illegible]

$$\operatorname{rank}(sE_{22} - A_{22}) = \operatorname{rank}(s\mathcal{E}_{22} - \mathcal{A}_{22}) = n_2 \quad \forall s \in \mathbf{C}. \tag{44}$$

[illegible]

$$\mathcal{U} = \begin{matrix} \\ \end{matrix} \begin{matrix} n_1 & n_2 \end{matrix} \begin{bmatrix} \mathcal{U}_{11} & \mathcal{U}_{12} \\ \mathcal{U}_{21} & \mathcal{U}_{22} \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \end{matrix}, \quad \mathcal{V} = \begin{matrix} \\ \end{matrix} \begin{matrix} n_1 & n_2 \end{matrix} \begin{bmatrix} \mathcal{V}_{11} & \mathcal{V}_{12} \\ \mathcal{V}_{21} & \mathcal{V}_{22} \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \end{matrix}, \tag{45}$$

$$\begin{bmatrix} \mathcal{U}_{11} & \mathcal{U}_{12} \\ 0 & I \end{bmatrix} UB = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \end{matrix}, \quad CV \begin{bmatrix} I & \mathcal{V}_{12} \\ 0 & \mathcal{V}_{22} \end{bmatrix} = \begin{matrix} n_1 & n_2 \\ \end{matrix} \begin{bmatrix} C_1 & C_2 \end{bmatrix}.$$

- [illegible] Q [illegible] $\mathcal{Q}$ [illegible]

$$\mathcal{Q}^T (sE_{22} - A_{22}) Q = \begin{matrix} \\ \end{matrix} \begin{matrix} \zeta_1 & \zeta_2 & \zeta_3 \end{matrix} \begin{bmatrix} s\Lambda_{11} - \Gamma_{11} & -\Gamma_{12} & -\Gamma_{13} \\ -\Gamma_{21} & 0 & 0 \\ -\Gamma_{31} & 0 & -\Gamma_{33} \end{bmatrix} \begin{matrix} \}\zeta_1 \\ \}\zeta_2 \\ \}\zeta_3 \end{matrix}, \tag{46}$$

[illegible] Λ_{11} [illegible] Γ_{33} [illegible]

- [illegible] [33, 39],

$$\left(\begin{bmatrix} \mathcal{U}_{11} E_{11} & 0 \\ 0 & (\mathcal{U}_{11} E_{11})^T \end{bmatrix}; \begin{bmatrix} \mathcal{U}_{11} A_{11} & 0 \\ 0 & -(\mathcal{U}_{11} A_{11})^T \end{bmatrix}, \begin{bmatrix} B_1 \\ C_1^T \end{bmatrix} \right)$$

$$
(47)\qquad \begin{cases} \mathcal{P}\begin{bmatrix} s\mathcal{U}_{11}E_{11}-\mathcal{U}_{11}A_{11} & 0 \\ 0 & s(\mathcal{U}_{11}E_{11})^T+(\mathcal{U}_{11}A_{11})^T \end{bmatrix} P \\ \qquad\qquad \begin{matrix} \nu_1 & \nu_2 \end{matrix} \\ \qquad = \begin{bmatrix} s\Theta_{11}-\Phi_{11} & s\Theta_{12}-\Phi_{12} \\ 0 & s\Theta_{22}-\Phi_{22} \end{bmatrix} \begin{matrix} \}\nu_1 \\ \}\nu_2 \end{matrix}, \\ \mathcal{P}\begin{bmatrix} B_1 \\ C_1^T \end{bmatrix} = \begin{bmatrix} \Psi_1 \\ 0 \end{bmatrix} \begin{matrix} \}\nu_1 \\ \}\nu_2 \end{matrix}, \end{cases}
$$

P $\mathcal{P}$ $(\Theta_{11};\Phi_{11},\Psi_1)$

$$
(48)\qquad \begin{bmatrix} C_1 & -B_1^T \end{bmatrix} P = \begin{matrix} \nu_1 & \nu_2 \end{matrix} \begin{bmatrix} \mathcal{K}_1 & \mathcal{K}_2 \end{bmatrix}, \quad P = \begin{matrix} \nu_1 & \nu_2 \end{matrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \end{matrix}.
$$

(i) $\mathcal{U}_{11}$ $\mathcal{V}_{22}$ $\zeta_2 \le \zeta_1$

(ii) (1)

$$
(49)\qquad \mathcal{K}_1 = 0, \quad P_{11}^T(\mathcal{U}_{11}E_{11})^T P_{21} \ge 0,
$$

$$
(50)\qquad \operatorname{rank}\begin{bmatrix} A_{22}\mathcal{V}_{22} & 0 & B_2 \\ 0 & (A_{22}\mathcal{V}_{22})^T & C_2^T \\ C_2 & B_2^T & D+D^T \end{bmatrix} = 2n_2,
$$

$$
(51)\qquad \operatorname{rank}\begin{bmatrix} A_{22} & 0 & 0 & 0 & B_2 \\ -E_{22} & A_{22}\mathcal{V}_{22} & 0 & 0 & 0 \\ 0 & 0 & A_{22}^T & -E_{22}^T & 0 \\ 0 & 0 & 0 & (A_{22}\mathcal{V}_{22})^T & C_2^T \\ 0 & C_2 & -B_2^T & 0 & 0 \end{bmatrix} = 4n_2,
$$

$$
(52)\qquad \operatorname{rank}\begin{bmatrix} A_{22} & 0 & B_2 \\ -E_{22} & A_{22}\mathcal{V}_{22} & 0 \\ 0 & C_2 & sI \end{bmatrix} = 2n_2+m \quad \forall s \in \mathbf{C}_+,
$$

$$
(53)\qquad \zeta_1 = \zeta_2.
$$

(i) First, a simple calculation yields that

$$
EV\begin{bmatrix} \mathcal{V}_{12} \\ \mathcal{V}_{22} \end{bmatrix} = AV\begin{bmatrix} \mathcal{V}_{12} \\ \mathcal{V}_{22} \end{bmatrix} \mathcal{A}_{22}^{-T}\mathcal{E}_{22}^T,
$$

which with the nonsingularity of $\mathcal{E}_{11}$ and the property (44) implies that the columns of $V\left[\begin{smallmatrix}\mathcal{V}_{12}\\ \mathcal{V}_{22}\end{smallmatrix}\right]$ form a basis of the eigenspace of the pencil $sE-A$ corresponding to all its infinite generalized eigenvalues [19]. Next, since all generalized eigenvalues of

$sE_{11}-A_{11}$ are finite and all generalized eigenvalues of $sE_{22}-A_{22}$ are infinite, therefore there exist unique $X \in \mathbf{R}^{n_1\times n_2}$ and $Y \in \mathbf{R}^{n_2\times n_1}$ such that

$$\begin{bmatrix} I & X \\ 0 & I \end{bmatrix} U(sE-A)V \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix} = \begin{bmatrix} sE_{11}-A_{11} & 0 \\ 0 & sE_{22}-A_{22} \end{bmatrix}.$$

This means that the columns of $V\left[\begin{smallmatrix} Y \\ I \end{smallmatrix}\right]$ also form a basis of the eigenspace of the pencil $sE-A$ corresponding to all its infinite generalized eigenvalues. Thus,

$$\begin{bmatrix} Y \\ I \end{bmatrix} = \begin{bmatrix} \mathcal{V}_{12} \\ \mathcal{V}_{22} \end{bmatrix} \mathcal{Z}$$

for some nonsingular $\mathcal{Z} \in \mathbf{R}^{n_2\times n_2}$, and so $\mathcal{V}_{22}$ is nonsingular. Similarly, we can show that $\mathcal{U}_{11}$ is also nonsingular.

Note that $sE_{22}-A_{22}$ is nonsingular for all $s \in \mathbf{C}$, and therefore ζ_1 and ζ_2 in (46) must satisfy

$$\zeta_2 \le \zeta_1.$$

(ii) Since

$$\begin{aligned} &\begin{bmatrix} \mathcal{U}_{11} & \mathcal{U}_{12} \\ 0 & I \end{bmatrix} U(sE-A)V \begin{bmatrix} I & \mathcal{V}_{12} \\ 0 & \mathcal{V}_{22} \end{bmatrix} \\ &\qquad = \begin{bmatrix} s\mathcal{U}_{11}E_{11}-\mathcal{U}_{11}A_{11} & 0 \\ 0 & sE_{22}\mathcal{V}_{22}-A_{22}\mathcal{V}_{22} \end{bmatrix}, \end{aligned} \tag{54}$$

we have

$$\begin{aligned} G(s) &= D + C(sE-A)^{-1}B \\ &= D + C_1(s\mathcal{U}_{11}E_{11}-\mathcal{U}_{11}A_{11})^{-1}B_1 + C_2(sE_{22}\mathcal{V}_{22}-A_{22}\mathcal{V}_{22})^{-1}B_2 \\ &= D + C_1(sI-(\mathcal{U}_{11}E_{11})^{-1}\mathcal{U}_{11}A_{11})^{-1}(\mathcal{U}_{11}E_{11})^{-1}B_1 \\ &\quad - C_2(I-s(A_{22}\mathcal{V}_{22})^{-1}E_{22}\mathcal{V}_{22})^{-1}(A_{22}\mathcal{V}_{22})^{-1}B_2 \\ &= G_1(s) - sC_2\mathcal{V}_{22}^{-1}A_{22}^{-1}E_{22}A_{22}^{-1}B_2 \\ &\quad - \sum_{k=2}^{n_2-1} s^k C_2((A_{22}\mathcal{V}_{22})^{-1}E_{22}\mathcal{V}_{22})^k(A_{22}\mathcal{V}_{22})^{-1}B_2, \end{aligned}$$

where

$$G_1(s) = D - C_2(A_{22}\mathcal{V}_{22})^{-1}B_2 + C_1(sI-(E_{11})^{-1}A_{11})^{-1}(\mathcal{U}_{11}E_{11})^{-1}B_1.$$

Consequently, we obtain using Lemma 10 that $G(s)$ is lossless positive real if and only if the following conditions hold:

(a) $G_1(s)$ is lossless positive real;

(b) $C_2\mathcal{V}_{22}^{-1}A_{22}^{-1}E_{22}A_{22}^{-1}B_2 \le 0$, i.e.,

$$\begin{cases} C_2\mathcal{V}_{22}^{-1}A_{22}^{-1}E_{22}A_{22}^{-1}B_2 = (C_2\mathcal{V}_{22}^{-1}A_{22}^{-1}E_{22}A_{22}^{-1}B_2)^T, \\ \operatorname{rank}(sI - C_2\mathcal{V}_{22}^{-1}A_{22}^{-1}E_{22}A_{22}^{-1}B_2) = m \quad \forall s \in \mathbf{C}_+; \end{cases}$$

(c) $C_2((A_{22}\mathcal{V}_{22})^{-1}E_{22}\mathcal{V}_{22})^k(A_{22}\mathcal{V}_{22})^{-1}B_2 = 0,\ i = 2, \ldots, n_2 - 1.$

Regarding (a), (b), and (c) above, we observe the following:

(1) Equations (47) and (48) can be rewritten as

$$\begin{cases} \left(\begin{bmatrix} \Theta_{11} & \Theta_{12} \\ 0 & \Theta_{22} \end{bmatrix}^{-1} \mathcal{P} \begin{bmatrix} \mathcal{U}_{11}E_{11} & 0 \\ 0 & I \end{bmatrix} \right)^{-1} = \begin{bmatrix} I & 0 \\ 0 & (\mathcal{U}_{11}E_{11})^T \end{bmatrix} P \\ \qquad = \begin{bmatrix} P_{11} & P_{12} \\ (\mathcal{U}_{11}E_{11})^T P_{21} & (\mathcal{U}_{11}E_{11})^T P_{22} \end{bmatrix}, \\ \left(\begin{bmatrix} I & 0 \\ 0 & (\mathcal{U}_{11}E_{11})^T \end{bmatrix} P \right)^{-1} \begin{bmatrix} E_{11}^{-1}A_{11} & 0 \\ 0 & -A_{11}^T(E_{11})^{-T} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & (\mathcal{U}_{11}E_{11})^T \end{bmatrix} P \\ \qquad = \begin{bmatrix} \Theta_{11}^{-1}\Phi_{11} & \Theta_{11}^{-1}\Phi_{12} - \Theta_{11}^{-1}\Theta_{12}\Theta_{22}^{-1}\Phi_{22} \\ 0 & \Theta_{22}^{-1}\Phi_{22} \end{bmatrix}, \\ \left(\begin{bmatrix} I & 0 \\ 0 & (\mathcal{U}_{11}E_{11})^T \end{bmatrix} P \right)^{-1} \begin{bmatrix} (\mathcal{U}_{11}E_{11})^{-1}B_1 \\ C_1^T \end{bmatrix} = \begin{bmatrix} \Theta_{11}^{-1}\Psi_1 \\ 0 \end{bmatrix} \end{cases}$$

and

$$\begin{bmatrix} C_1 & -B_1^T(\mathcal{U}_{11}E_{11})^{-T} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & (\mathcal{U}_{11}E_{11})^T \end{bmatrix} P = \begin{bmatrix} C_1 & -B_1^T \end{bmatrix} P = \begin{bmatrix} \mathcal{K}_1 & \mathcal{K}_2 \end{bmatrix},$$

respectively. By (45) and (54), the controllability of $(E; A, B)$ and $(E^T; A^T, C^T)$ implies the controllability of $(E_{11}^{-1}A_{11}, (\mathcal{U}_{11}E_{11})^{-1}B_1)$ and $((E_{11}^{-1}A_{11})^T, C_1^T)$. According to Theorem 16, $G_1(s)$ is lossless positive real if and only if

$$\mathcal{K}_1 = 0, \quad P_{11}^T(\mathcal{U}_{11}E_{11})^T P_{21} \ge 0,$$

and

$$(D - C_2(A_{22}\mathcal{V}_{22})^{-1}B_2) + (D - C_2(A_{22}\mathcal{V}_{22})^{-1}B_2)^T = 0,$$

$$\text{i.e., } \operatorname{rank} \begin{bmatrix} A_{22}\mathcal{V}_{22} & 0 & B_2 \\ 0 & (A_{22}\mathcal{V}_{22})^T & C_2^T \\ C_2 & B_2^T & D + D^T \end{bmatrix} = 2n_2,$$

equivalently, if and only if the conditions (49) and (50) hold.

(2) It is also easy to see that

$$C_2\mathcal{V}_{22}^{-1}A_{22}^{-1}E_{22}A_{22}^{-1}B_2 \le 0$$

if and only if the conditions (51) and (52) hold.

(3) Since $sE_{22}-A_{22}$ is nonsingular for all $s \in \mathbf{C}$, there exists a nonsingular matrix Z [17] such that

$$Z^{-1}(\mathcal{V}_{22}^{-1}A_{22}^{-1}E_{22}\mathcal{V}_{22})Z = \begin{array}{c} \begin{matrix} \tau_1 & \tau_2 & \tau_3 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{matrix} \}\tau_1 \\ \}\tau_3 \\ \}\tau_2 \end{matrix}, \quad \tau_2 \le \tau_3. \tag{55}$$

By (45) and (54) again, the controllability of $(E; A, B)$ and $(E^T; A^T, C^T)$ implies

$$\operatorname{rank}\begin{bmatrix} \alpha(A_{22}\mathcal{V}_{22})^{-1}E_{22}\mathcal{V}_{22} - \beta I & (A_{22}\mathcal{V}_{22})^{-1}B_2 \end{bmatrix}$$

$$= \begin{bmatrix} \alpha(A_{22}\mathcal{V}_{22})^{-1}E_{22}\mathcal{V}_{22} - \beta I \\ C_2 \end{bmatrix} = n_2$$

for all $(\alpha, \beta) \in \mathbf{C}^2\backslash\{0,0\}$. Hence, we have using Lemma 11 that $C_2((A_{22}\mathcal{V}_{22})^{-1}E_{22}\mathcal{V}_{22})^k(A_{22}\mathcal{V}_{22})^{-1}B_2 = 0$ $(k = 2, \ldots, n_2 - 1)$ if and only if

$$\tau_2 = \tau_3. \tag{56}$$

To find the relation between the conditions (53) and (56), let us refine the factorization (46). Since A_{22} is nonsingular, therefore Γ_{12} and Γ_{21} in (46) are of full column rank and full row rank, respectively, and

$$\zeta_2 \le \zeta_1. \tag{57}$$

Note that A_{22}, Γ_{33}, and Λ_{11} are nonsingular, Γ_{12} is of full column rank, and Γ_{21} is of full row rank. It is easy to see that there exist nonsingular matrices $\mathcal{Y}_1$ and $\mathcal{Y}_2$ such that

$$\mathcal{Y}_1\mathcal{Q}^T A_{22}Q\mathcal{Y}_2 = \begin{array}{c} \begin{matrix} \zeta_2 & \zeta_1-\zeta_2 & \zeta_2 & \zeta_3 \end{matrix} \\ \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & I & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \end{array} \begin{matrix} \}\zeta_2 \\ \}\zeta_1-\zeta_2 \\ \}\zeta_2 \\ \}\zeta_3 \end{matrix},$$

$$\mathcal{Y}_1\mathcal{Q}^T E_{22}Q\mathcal{Y}_2 = \begin{array}{c} \begin{matrix} \zeta_2 & \zeta_1-\zeta_2 & \zeta_2 & \zeta_3 \end{matrix} \\ \begin{bmatrix} \Lambda_{11}^{(1,1)} & \Lambda_{11}^{(1,2)} & 0 & 0 \\ \Lambda_{11}^{(2,1)} & \Lambda_{11}^{(2,2)} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{matrix} \}\zeta_2 \\ \}\zeta_1-\zeta_2 \\ \}\zeta_2 \\ \}\zeta_3 \end{matrix},$$

where

$$\operatorname{rank}\begin{bmatrix} \Lambda_{11}^{(1,1)} & \Lambda_{11}^{(1,2)} \\ \Lambda_{11}^{(2,1)} & \Lambda_{11}^{(2,2)} \end{bmatrix} = \operatorname{rank}(\Lambda_{11}) = \zeta_1 = \zeta_2 + (\zeta_1 - \zeta_2). \tag{58}$$

Consequently, there exist permutation matrices $\mathcal{Y}_3$ and $\mathcal{Y}_4$ such that

$$\mathcal{Y}_3\mathcal{Y}_1\mathcal{Q}^T A_{22}Q\mathcal{Y}_2\mathcal{Y}_4 = \begin{array}{c} \begin{matrix} \zeta_3 & \zeta_2 & \zeta_1-\zeta_2 & \zeta_2 \end{matrix} \\ \left[\begin{array}{cccc} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{array}\right] \end{array} \begin{matrix} \}\zeta_3 \\ \}\zeta_2 \\ \}\zeta_1-\zeta_2 \\ \}\zeta_2 \end{matrix}$$

and

$$\mathcal{Y}_3\mathcal{Y}_1\mathcal{Q}^T E_{22}Q\mathcal{Y}_2\mathcal{Y}_4 = \begin{array}{c} \begin{matrix} \zeta_3 & \zeta_2 & \zeta_1-\zeta_2 & \zeta_2 \end{matrix} \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & \Lambda_{11}^{(1,2)} & \Lambda_{11}^{(1,1)} \\ 0 & 0 & \Lambda_{11}^{(2,2)} & \Lambda_{11}^{(2,1)} \\ 0 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{matrix} \}\zeta_3 \\ \}\zeta_2 \\ \}\zeta_1-\zeta_2 \\ \}\zeta_2 \end{matrix},$$

which gives

(59)

$$(\mathcal{Y}_2\mathcal{Y}_4)^{-1}(Q^{-1}A_{22}^{-1}E_{22}Q)(\mathcal{Y}_2\mathcal{Y}_4) = \begin{array}{c} \begin{matrix} \zeta_3 & \zeta_2 & \zeta_1-\zeta_2 & \zeta_2 \end{matrix} \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & \Lambda_{11}^{(1,2)} & \Lambda_{11}^{(1,1)} \\ 0 & 0 & \Lambda_{11}^{(2,2)} & \Lambda_{11}^{(2,1)} \\ 0 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{matrix} \}\zeta_3 \\ \}\zeta_2 \\ \}\zeta_1-\zeta_2 \\ \}\zeta_2 \end{matrix}.$$

By a simple comparison between (59) and (55), we know by taking (57) and (58) into account that

$$\zeta_3 = \tau_1, \quad \zeta_2 = \tau_2, \quad \zeta_1 = \tau_3.$$

Thus, (56) holds if and only if (53) holds.

Therefore, Theorem 17 follows. □

2. The condition (52) is equivalent to stating that the pencil

$$\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}, \begin{bmatrix} -A_{22} & 0 & -B_2 \\ E_{22} & -A_{22}\mathcal{V}_{22} & 0 \\ 0 & -C_2 & 0 \end{bmatrix}\right)$$

has no finite generalized eigenvalues on $\mathbf{C}_+$. Hence, it can be verified easily, as follows:

- Compute the finite generalized eigenvalues of the pencil

$$\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{bmatrix}, \begin{bmatrix} -A_{22} & 0 & -B_2 \\ E_{22} & -A_{22}\mathcal{V}_{22} & 0 \\ 0 & -C_2 & 0 \end{bmatrix}\right).$$

  If such finite generalized eigenvalues are all on $\mathbf{C}\backslash\mathbf{C}^+$, then the condition (52) holds; otherwise, it does not hold.

3. The form (46) can be computed easily, as follows:

- Compute the SVD of E_{22} [19] to get orthogonal matrices Q_1 and $\mathcal{Q}_1$ such that

$$\mathcal{Q}_1^T E_{22} Q_1 = \begin{array}{c} \begin{matrix} \zeta_1 & n_2 - \zeta_1 \end{matrix} \\ \begin{bmatrix} \Lambda_{11} & 0 \\ 0 & 0 \end{bmatrix} \end{array} \begin{matrix} \}\zeta_1 \\ \}n_2 - \zeta_1 \end{matrix},$$

where Λ_{11} is nonsingular. Set

$$\mathcal{Q}_1^T A_{22} Q_1 = \begin{array}{c} \begin{matrix} \zeta_1 & n_2 - \zeta_1 \end{matrix} \\ \begin{bmatrix} \Gamma_{11} & \Gamma_{12}^{(1)} \\ \Gamma_{21}^{(1)} & \Gamma_{22}^{(1)} \end{bmatrix} \end{array} \begin{matrix} \}\zeta_1 \\ \}n_2 - \zeta_1 \end{matrix}.$$

- Compute the SVD of $\Gamma_{22}^{(1)}$ to get orthogonal matrices Q_2 and $\mathcal{Q}_2$ such that

$$\mathcal{Q}_2^T \Gamma_{22}^{(1)} Q_2 = \begin{array}{c} \begin{matrix} \zeta_2 & \zeta_3 \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & \Gamma_{33} \end{bmatrix} \end{array} \begin{matrix} \}\zeta_2 \\ \}\zeta_3 \end{matrix},$$

where Γ_{33} is nonsingular. Set

$$\Gamma_{12}^{(1)} Q_2 = \begin{array}{c} \begin{matrix} \zeta_2 & \zeta_3 \end{matrix} \\ \begin{bmatrix} \Gamma_{12} & \Gamma_{13} \end{bmatrix} \end{array}, \quad \mathcal{Q}_2^T \Gamma_{21}^{(1)} = \begin{array}{c} \\ \begin{bmatrix} \Gamma_{21} \\ \Gamma_{31} \end{bmatrix} \end{array} \begin{matrix} \\ \}\zeta_2 \\ \}\zeta_3 \end{matrix}$$

and

$$Q = Q_1 \begin{bmatrix} I & 0 \\ 0 & Q_2 \end{bmatrix}, \quad \mathcal{Q} = \mathcal{Q}_1 \begin{bmatrix} I & 0 \\ 0 & \mathcal{Q}_2 \end{bmatrix}.$$

Then $\mathcal{Q}^T(sE_{22} - A_{22})Q$ is in the form (46).

3. Conclusions. We have studied the algebraic characterizations for the positive realness of descriptor systems in this paper. The main contributions of the present work are as follows:

- In Theorem 13 we have algebraically characterized the positive realness of descriptor systems by using a linear matrix inequality of the form (14) and thus extended the well-known positive real lemma for standard state space systems to descriptor systems.
- We have also shown in Theorems 16 and 17 that the lossless positive realness of standard state space systems and descriptor systems can be tested easily using the controllable staircase forms of the standard state space systems and the generalized controllable staircase forms of descriptor systems, respectively.

Acknowledgments. We are grateful to the anonymous referees and the associate editor Professor Peter Benner for their valuable comments and suggestions.

REFERENCES

[1] B. D. O. Anderson, *A system theory criterion for positive real matrices*, SIAM J. Control, 5 (1967), pp. 171–182.

[2] B. D. O. Anderson, M. Mansour, and F. J. Kraus, *A new test for strict positive realness*, IEEE Trans. Circuits Systems I Fund. Theory Appl., 42 (1995), pp. 226–229.

[3] B. D. O. Anderson and S. Vongpanitlerd, *Network Analysis and Synthesis*, Prentice–Hall, Englewood Cliffs, NJ, 1973.

[4] K. J. Astrom, *Theory and applications of adaptive control—A survey*, Automatica J. IFAC, 9 (1983), pp. 471–481.

[5] Z. Bai and R. W. Freund, *Eigenvalue-based characterization and test for positive realness of scale transfer functions*, IEEE Trans. Automat. Control, AC-45 (2000), pp. 2396–2401.

[6] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, PA, 1994.

[7] O. Brune, *Synthesis of a finite two-terminal network whose driving-point impedance is a prescribed function of frequency*, J. Math. Phys., 10 (1931), pp. 191–236.

[8] B. M. Chen, Z. Lin, and Y. Shamash, *Linear Systems Theory: A Structural Decomposition Approach*, Birkhäuser, Boston, 2004.

[9] D. Chu and Y. S. Hung, *A numerical solution for the simultaneous disturbance rejection and row by row decoupling problem*, Linear Algebra Appl., 320 (2000), pp. 37–49.

[10] D. Chu and V. Mehrmann, *Disturbance decoupling for linear time-invariant systems: A matrix pencil approach*, IEEE Trans. Automat. Control, 46 (2001), pp. 802–808.

[11] D. Chu and D. D. Siljak, *A canonical form for the inclusion principle of dynamic systems*, SIAM J. Control Optim., 44 (2005), pp. 969–990.

[12] L. Dai, *Singular Control Systems*, Springer-Verlag, Berlin, 1989.

[13] J. W. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil $A-\lambda B$: Robust software with error bounds and applications. Part* I: *Theory and algorithms*, ACM Trans. Math. Software, 19 (1993), pp. 160–174.

[14] C. A. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties*, Academic Press, New York, 1975.

[15] R. W. Freund, *Krylov-subspace methods for reduced-order modeling in circuit simulation*, J. Comput. Appl. Math., 123 (2000), pp. 395–421.

[16] R. W. Freund and F. Jarre, *An extension of the positive real lemma to descriptor systems*, Optim. Methods Softw., 19 (2004), pp. 69–87.

[17] F. Gantmacher, *The Theory of Matrices*, Vol. 2, Chelsea, New York, 1959.

[18] Y. Genin, Y. Nesterov, R. Stefan, P. Van Dooren, and S. Xu, *Positivity and linear matrix inequality*, Eur. J. Control, 8 (2002), pp. 275–298.

[19] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[20] M. He and B. M. Chen, *Structural decomposition of linear singular systems: The single-input and single-output case*, Systems Control Lett., 47 (2002), pp. 325–332.

[21] M. He, B. M. Chen, and Z. Lin, *Structural decomposition and its properties of general multivariable linear singular systems*, in Proceedings of the 2003 American Control Conference, Denver, CO, American Control Society, 2003, pp. 4494–4499.

[22] I. Hodaka, N. Sakamoto, and M. Suzuki, *New results for strict positive realness and feedback stability*, IEEE Trans. Automat. Control, AC-45 (2000), pp. 813–819.

[23] P. Ioannou and G. Tao, *Frequency domain conditions for strictly positive real functions*, IEEE Trans. Automat. Control, AC-32 (1987), pp. 53–54.

[24] T. Iwasaki and S. Hara, *Generalized KYP lemma: Unified frequency domain inequalities with design applications*, IEEE Trans. Automat. Control, AC-50 (2005), pp. 41–59.

[25] V. A. Jakubovic, *The solution of certain matrix inequalities in automatic control theory*, Soviet Math. Dokl., 3 (1962), pp. 620–623.

[26] R. E. Kalman, *Lyapunov functions for the problem of Lur'e in automatic control*, Proc. Natl. Acad. Sci. USA, 49 (1963), pp. 201–205.

[27] R. E. Kalman, *On a new characterization of linear passive systems*, in Proceedings of the First Annual Allerton Conference on Circuit and System Theory, University of Illinois, 1963, pp. 456–470.

[28] Y. D. Landau, *Adaptive Control—The Model Reference Approach*, Marcel Dekker, New York, 1979.

[29] R. Lozano-Leal and S. M. Joshi, *Strictly positive real transfer functions revisited*, IEEE Trans. Automat. Control, AC-35 (1990), pp. 1243–1245.

[30] H. J. Marquez and C. J. Damaren, *On the design of strictly positive real transfer functions*,

IEEE Trans. Automat. Control, AC-42 (1995), pp. 214–218.
[31] I. Masubuchi, *Dissipativity inequality for continuous-time descriptor systems: A realization-independent condition*, in Proceedings of the IFAC Symposium on Large Scale Systems, 2004, pp. 417–420.
[32] I. Masubuchi, *Dissipativity inequalities for continuous-time descriptor systems with applications to synthesis of control gains*, Systems Control Lett., 55 (2006), pp. 158–164.
[33] G. S. Miminis, *Deflation in eigenvalue assignment of descriptor systems using state feedback*, IEEE Trans. Automat. Control, AC-38 (1993), pp. 1322–1336.
[34] T. Mita, Y. Chida, and J. W. Wang, *Strictly positive real condition and pseudo-strictly positive real condition*, Trans. SICE, 25 (1989), pp. 751–757.
[35] V. Popov, *Absolute stability of nonlinear systems of automatic control*, Automat. Remote Control, 22 (1962), pp. 857–875.
[36] V. Popov, *Hyperstability of Control Systems*, Springer-Verlag, Berlin, 1973.
[37] D. D. Siljak, *New algebraic criteria for positive realness*, J. Franklin Inst., 291 (1971), pp. 109–120.
[38] G. Tao and P. Ioannou, *Strictly positive real matrices and the Lefschetz-Kalman-Yakubovich lemma*, IEEE Trans. Automat. Control, AC-33 (1988), pp. 1183–1185.
[39] P. Van Dooren, *The generalized eigenstructure problem in linear system theory*, IEEE Trans. Automat. Control, AC-26 (1981), pp. 111–129.
[40] H.-S. Wang and F.-R. Chang, *The generalized state-space description of positive realness and bounded realness*, in Proceedings of the 39th IEEE Midwest Symposium on Circuits and System, 1997, pp. 893–896.
[41] H. Weiss, Q. Wang, and J. L. Speyer, *System characterization of positive real conditions*, IEEE Trans. Automat. Control, AC-39 (1994), pp. 541–544.
[42] J. T. Wen, *Time domain and frequency domain conditions for strict positive realness*, IEEE Trans. Automat. Control, AC-33 (1988), pp. 988–992.
[43] W. M. Wonham, *Linear Multivariable Control: A Geometric Approach*, 2nd ed., Springer-Verlag, New York, 1985.
[44] C. Xiao and D. J. Hill, *Concepts of strict positive realness and the absolute stability problem of continuous-time systems*, Automatica, 34 (1998), pp. 1071–1082.
[45] L. Zhang, J. Lam, and S. Xu, *On positive realness of descriptor systems*, IEEE Trans. Circuits Systems I Fund. Theory Appl., 49 (2002), pp. 401–407.
[46] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice–Hall, Upper Saddle River, NJ, 1996.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 223–235

ON NUMERICAL ISSUES OF INTERIOR POINT METHODS*

CSABA MÉSZÁROS†

Abstract. This paper concerns some numerical stability issues of factorizations in interior point methods. In our investigation we focus on regularization techniques for the augmented system. We derive the fundamental property of regularization and necessary conditions for the convergence of iterative refinement. A relaxation technique is described that improves on convergence properties. We introduce a practical, adaptive technique to determine the required amount of regularization in numerically difficult situations. Numerical experiments on large-scale, numerically difficult linear programming problems are presented.

Key words. regularization, indefinite Cholesky factorization, interior point methods

AMS subject classifications. 90C51, 65F22, 65F05

DOI. 10.1137/050633354

1. Introduction. During the past 15 years, interior point methods (IPMs) proved to be efficient in practice and numerically robust for solving large-scale optimization problems [7, 1]. The important practical issue, the stability of the computations in IPMs, deserved special attention in the literature [6, 24, 27, 26]. For most interior point algorithms, the major computational task is to solve symmetric systems of linear equations, which is usually done by factorization in practice. One of the most important difficulties for IPMs is the ill-conditioning of these linear systems when the method approaches the optimal solution of the optimization problem. It has been shown that, in general, degeneracy in the optimization problem causes ill-conditioning, but in such a case the possible numerical errors appear to be in a subspace [17]. This situation can be handled well with a modified factorization scheme by skipping columns corresponding to small pivots during numerical computations [27, 17]. In [17] it was pointed out that ill-conditioning may also appear when the optimal solution of the problem is scaled badly, i.e., if the values of the strictly positive components in the optimal solution vector are of different orders of magnitude. A situation like this may easily occur in real-life optimization problems, resulting in a breakdown of the modified factorization scheme, as demonstrated in [17]. In such a case, a regularization technique can help to avoid numerical problems. In the present paper regularization means that prior to the symmetric factorizations, some "small" diagonal matrix values are replaced by "reasonably large" ones. Naturally, regularization is very sensitive to the manner of how the "small" values are identified and of how the "reasonably large" values are set.

In this paper we investigate regularization techniques. We introduce the practical linear algebra operations of IPMs in section 2. In section 3 we introduce a regularization scheme for the diagonal scaling matrix of IPMs and derive necessary and sufficient conditions for the convergence of iterative refinement on the regularized system. In section 4 we discuss special cases and describe a relaxation technique to improve the

*Received by the editors June 10, 2005; accepted for publication (in revised form) by L. Vandenberghe August 22, 2007; published electronically February 27, 2008. This work was supported by the Hungarian Research Fund OTKA T-043276.
http://www.siam.org/journals/simax/30-1/63335.html
†Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary (meszaros@sztaki.hu).

convergence of iterative refinement. In section 5 an adaptive scheme for the use of regularization is described. Section 6 presents numerical results.

2. Interior point methods and symmetric factorizations. We consider the linear programming problem and a primal-dual log barrier interior point method for further investigations. It is to be noted that the underlying linear algebra of other interior point approaches is fairly similar.

Let us consider the linear programming problem

$$\begin{array}{rcl} \min\ c^T x, & & \\ Ax & = & b, \\ x & \geq & 0, \end{array} \tag{2.1}$$

where $x, c \in \mathcal{R}^n$, $A \in \mathcal{R}^{m \times n}$ is of full row rank, and $b \in \mathcal{R}^m$. The associated dual problem is

$$\begin{aligned} \max \quad & b^T y, \\ & A^T y + z = c, \\ & z \geq 0, \end{aligned}$$

where $y \in R^m$ and $z \in R^n$. The logarithmic barrier problem corresponding to (2.1) is

$$\begin{aligned} \min\ & c^T x - \mu \sum_{i=1}^{n} \ln\, x_i, \\ & Ax = b, \quad x > 0, \end{aligned} \tag{2.2}$$

where μ is a positive scalar barrier parameter. A log barrier IPM approaches the optimal solution on the central path (x^*, y^*, z^*) by a sequence of barrier problems (2.2), while the barrier parameter is decreased toward zero. Following the classical introduction of the primal-dual log barrier method [12, 8, 25], the algorithm can be derived by applying Newton's method to solve the Karush–Kuhn–Tucker system of (2.2). Present efficient implementations employ the predictor-corrector and higher-order correction techniques [13, 1].

The computational task in each iteration of the resulting methods is the solution of systems of linear equations

$$\begin{bmatrix} D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{2.3}$$

where

$$D = \operatorname{diag}\left(x_i^{-1} z_i\right).$$

Later, the matrix of (2.3) is referred to as the augmented system. For the predictor-corrector primal-dual method, two systems should be solved with the same matrix and different right-hand sides, while higher-order methods require more solutions of systems like the above.

In practice, a Cholesky-like decomposition of the permuted matrix is computed as

$$L \Lambda L^T = P \begin{bmatrix} D & A^T \\ A & 0 \end{bmatrix} P^T, \tag{2.4}$$

where L is lower triangular, Λ is diagonal with both positive and negative values, and P is a permutation. Implementations either use the normal equations, for which

$$L = \begin{bmatrix} D^{\frac{1}{2}} & \\ AD^{-\frac{1}{2}} & \hat{L} \end{bmatrix}, \quad \Lambda = \begin{bmatrix} I & \\ & -I \end{bmatrix}, \quad P = \begin{bmatrix} I & \\ & \hat{P} \end{bmatrix},$$

where $\hat{L}\hat{L}^T$ is the Cholesky decomposition of $AD^{-1}A^T$ [20, 1], or alter the matrix to be quasi-definite, so it can be factorized with any P by indefinite Cholesky factorization [23, 22]. Our implementation uses a heuristic to select one of these two approaches. The indefinite Cholesky factorization is used only if a reduction in the fill-in seems possible [10]. From now on we will omit the permutation P, as it has no influence on the algebraic properties.

During interior point iterations the diagonal values of D converge to zero or to infinity, depending on the analytic center x^* of the optimal face of problem (2.1):

$$D_{ii} \to \begin{cases} 0 & \text{if } x_i^* > 0, \\ \infty & \text{if } x_i^* = 0, \end{cases}$$

resulting in increasing ill-conditioning of the augmented system. Supposing that the original constraint matrix A is well-conditioned, in most cases ill-conditioning like this presents no numerical difficulties, except if the solution x^* is scaled badly; i.e., its positive components have a significantly different order of magnitude [17]. We investigate this situation where the usual factorization approaches may break down.

In interior point implementations, iterative refinement is a standard technique for improving numerical accuracy. The classical normal equations approach, where (2.3) is reduced to

$$-AD^{-1}A^T\Delta y = \beta - D^{-1}A^T\alpha \tag{2.5}$$

and (2.5) is solved for Δy, is very attractive because the positive definiteness of the matrix $AD^{-1}A^T$ allows powerful iterative techniques, such as the preconditioned conjugate gradient method. But we have observed that iterative methods for (2.5) often fail in interior point implementations in numerically difficult cases. The reason is that the matrix $AD^{-1}A^T$ often cannot be formed accurately, so that Δy will be inaccurate, and then Δx will be inaccurately obtained from $D\Delta x = \alpha - A^T\Delta y$. Therefore, we apply iterative refinement on the augmented system, which feeds back corrections corresponding to the residuals in the spaces of both Δx and Δy. Regarding the pure least squares problem, this approach was justified as well by theory [3, 2].

The steps of a refinement scheme like this can be written as

$$\begin{aligned} &\text{solve} \quad \bar{M}\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^0 = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \\ &\text{solve} \quad \bar{M}\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} - M\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^k, \\ &\text{form} \quad \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^{k+1} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^k + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \end{aligned} \tag{2.6}$$

where

$$M = \begin{bmatrix} D & A^T \\ A & 0 \end{bmatrix}$$

and $\bar{M}$ is an approximation of M suitable for the solve operation. Later, we will consider iterative refinement in the following standard form:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^{k+1} = \left(I - \bar{M}^{-1}M\right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^{k} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}^{0}. \tag{2.7}$$

In the implementation we use the IEEE standard double precision arithmetic for all computation steps. Note that "iterative refinement" normally requires additional precision in computing the right-hand side of (6), but here we are solving systems involving a modified $\bar{M}$ instead of the required M. This is similar to using inaccurate factors from an unstable factorization of M. In such contexts, a few steps of "refinement" can be helpful even without extra precision.

3. Regularization. In choosing $\bar{M}$, our approach is to modify the diagonals of M. In this way we regularize D, the so-called diagonal scaling matrix of interior methods. Let us suppose that scaling factors different from D are used. Let $\bar{D}$ be a diagonal positive definite $n \times n$ matrix and consider the regularized augmented matrix

$$\bar{M} = \begin{bmatrix} \bar{D} & A^T \\ A & 0 \end{bmatrix}.$$

In our approach we factorize $\bar{M}$ and apply iterative refinement to achieve the solution of (2.3). In exact arithmetic, the iterative process converges if the eigenvalues λ of the refinement operator satisfy $\left|\lambda\left(I - \bar{M}^{-1}M\right)\right| < 1$ [4]. Thus we investigate the eigenvalues by examining the solutions of the equation

$$\det\left(I - \bar{M}^{-1}M - \lambda I\right) = 0. \tag{3.1}$$

Since $\bar{M}$ is nonsingular, (3.1) is equivalent to

$$\det\left(\bar{M}^{-1}\right) \det\left(M - (1-\lambda)\bar{M}\right) = 0.$$

The block matrix $M - (1-\lambda)\bar{M}$ can be simplified as

$$M - (1-\lambda)\bar{M} = \begin{bmatrix} D - (1-\lambda)\bar{D} & A^T - (1-\lambda)A^T \\ A - (1-\lambda)A & 0 \end{bmatrix} = \begin{bmatrix} D - (1-\lambda)\bar{D} & \lambda A^T \\ \lambda A & 0 \end{bmatrix}.$$

This shows that 0 is an eigenvalue with multiplicity of at least $2m$, since

$$\det\left(M - (1-\lambda)\bar{M}\right) = \lambda^{2m} \det \begin{bmatrix} D - (1-\lambda)\bar{D} & A^T \\ A & 0 \end{bmatrix}.$$

Thus, we can reduce (3.1) to

$$\det \begin{bmatrix} D - (1-\lambda)\bar{D} & A^T \\ A & 0 \end{bmatrix} = 0.$$

Now we state the main lemma.

LEMMA 3.1. [illegible] $\left(I - \bar{M}^{-1}M\right)$ [illegible]

[illegible] Let us suppose that there exists a complex eigenvalue: $\exists \lambda \in \mathcal{C} \setminus \mathcal{R}$. Since $\bar{D}_{ii} > 0$, the inverse of the diagonal complex matrix $(D - (1-\lambda)\bar{D})$ exists because its imaginary part is of full rank:

$$\exists \left(D - (1-\lambda)\bar{D}\right)^{-1} \in \mathcal{C}^{n \times n}.$$

We show that $A\left(D-(1-\lambda)\bar{D}\right)^{-1}A^T$ is of full rank, which contradicts the assumption. Since both λ and its complex conjugate $\bar{\lambda}$ are eigenvalues, we can assume that λ is of the form of $\lambda=\lambda_1-i\lambda_2$, where $\lambda_2>0$. Since for any $a,b\in\mathcal{R}$,

$$(a-ib)^{-1}=\frac{a+ib}{a^2+b^2},$$

the matrix $\left(D-(1-\lambda)\bar{D}\right)^{-1}$ can be written as

$$\left(D-(1-\lambda)\bar{D}\right)^{-1}=F+iG,$$

where F is diagonal and G is diagonal positive definite. Therefore,

$$A\left(D-(1-\lambda)\bar{D}\right)^{-1}A^T=AFA^T+iAGA^T,$$

and, furthermore,

$$\operatorname{rank}\left(AFA^T+iAGA^T\right)=\operatorname{rank}\left(\left(AGA^T\right)^{-1}AFA^T+iI\right).$$

The eigenvalues of $\left(AGA^T\right)^{-1}AFA^T$ are all real numbers because it is similar to a symmetric real matrix:

$$\left(AGA^T\right)^{-1}AFA^T\sim\left(AGA^T\right)^{-1/2}AFA^T\left(AGA^T\right)^{-1/2}.$$

This means that the eigenvalues of the matrix $\left(AGA^T\right)^{-1}AFA^T+iI$ have the form

$$\lambda\left(\left(AGA^T\right)^{-1}AFA^T+iI\right)=r+i,$$

where $r\in\mathcal{R}$. Therefore, no eigenvalue can be 0, because of the nonzero imaginary part. This shows that $A\left(D-(1-\lambda)\bar{D}\right)^{-1}A^T$ is of full rank for any $\lambda\in\mathcal{C}\setminus\mathcal{R}$. Thus, all eigenvalues of $\left(I-\bar{M}^{-1}M\right)$ should be in $\mathcal{R}$.

THEOREM 3.2. [illegible] $\left(I-\bar{M}^{-1}M\right)$ [illegible] 1 [illegible] For $\lambda\geq 1$, the matrix $D-(1-\lambda)\bar{D}$ is positive definite and

$$\begin{bmatrix} D-(1-\lambda)\bar{D} & A^T \\ A & 0 \end{bmatrix}$$

is of full rank. Therefore, for $\lambda\geq 1$ (3.1) cannot be satisfied.

THEOREM 3.3. [illegible] $\left(I-\bar{M}^{-1}M\right)$ [illegible] $\min_i\left(1-D_{ii}/\bar{D}_{ii}\right)$.

[illegible] For $\lambda<\min_i\left(1-D_{ii}/\bar{D}_{ii}\right)$ the matrix $D-(1-\lambda)\bar{D}$ is negative definite and

$$\begin{bmatrix} D-(1-\lambda)\bar{D} & A^T \\ A & 0 \end{bmatrix}$$

is of full rank.

COROLLARY 3.4. [illegible] $\left(I-\bar{M}^{-1}M\right)$ [illegible] -1.

$$\max\frac{D_{ii}}{\bar{D}_{ii}}<2. \tag{3.2}$$

We can now derive sufficient conditions for the convergence of iterative refine-

ment:

- the diagonals of D are increased arbitrarily, or
- the diagonals of D are decreased by less than 50%.

Note that the last condition is sharp, which can be demonstrated by the following example:

$$A = [1, 1], \quad D = \begin{bmatrix} 2 & \\ & 2 \end{bmatrix}, \quad \bar{D} = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}.$$

Iterative refinement does not converge because

$$\lambda\left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 0 \end{bmatrix}\right) = \{0,\ 0,\ -1\}.$$

4. Speeding up convergence. Let $R = I - \bar{M}^{-1}M$ denote the matrix of the iterative refinement (2.7). Observe that the following upper bound can be easily derived regarding the absolute value of the eigenvalues of R:

$$\max |\lambda(R)| \leq \max_i \left|1 - \frac{D_{ii}}{\bar{D}_{ii}}\right|. \tag{4.1}$$

This upper bound helps to improve the convergence rate of iterative refinement. Our idea is to use a relaxation technique and consider the matrix

$$\bar{M}_\gamma = \begin{bmatrix} \gamma\bar{D} & A^T \\ A & 0 \end{bmatrix}$$

and choose $\gamma > 0$ such that

$$\max_i \left|1 - \frac{D_{ii}}{\gamma\bar{D}_{ii}}\right|$$

is minimized. Then, $\bar{M}_\gamma$ is factored and used in the iterative refinement process (2.7).

The case $\gamma > 1$ is called overrelaxation and $\gamma < 1$ is underrelaxation. Since in our situation, usually $\bar{D}_{ii} \geq D_{ii}$, we may use underrelaxations only with $\gamma \leq 1$. In practice, we have $\bar{D}_{ii} = D_{ii}$ for several indices i, which requires that $\gamma > \frac{1}{2}$.

A relaxation technique like this can relax condition (3.2) since, for any $\bar{M}$, it is possible to find a large enough γ such that

$$\max \left|\lambda\left(I - \bar{M}_\gamma^{-1}M\right)\right| < 1.$$

This means that for any positive definite diagonal $\bar{D}$, we can determine a relaxation parameter γ such that iterative refinement will converge to the solution of (2.3).

In practice we determine γ by solving

$$\min_\gamma \sum_{i=1}^{n} \left(D_{ii} - \gamma\bar{D}_{ii}\right)^2. \tag{4.2}$$

In this way we reduce the Frobenius distance between the original and regularized scaling matrices.

4.1. Further properties of the regularization. Observe that if $m = n$, then R is nilpotent, i.e., $\exists k \leq m + n$ positive integer so that $R^k = 0$. Therefore, in exact arithmetic, iterative refinement converges in a finite number of steps. The same is true if the "regularized" columns of A are linearly independent of the other columns, i.e., if

$$D_{ii} \neq \bar{D}_{ii} \Longrightarrow a_i \notin \mathrm{Span}(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n).$$

Then, the matrix R is nilpotent, which again implies a finite number of steps for iterative refinement in exact arithmetic.

Next, we consider a regularization scheme similar to the Tyhonov–Levenberg regularization for LSQR [19], as used in [22]. We set

$$\bar{M} = \begin{bmatrix} \bar{D} & A^T \\ A & -D^+ \end{bmatrix}, \tag{4.3}$$

where $\bar{D}$ and D^+ are positive definite diagonal. The regularization D^+ corresponds to a regularization of the free variables in the dual problem [15]. We apply regularization like this to alter $\bar{M}$ to be quasi-definite for the indefinite Cholesky factorization in case the matrix of normal equations has significant fill [16]. For the behavior of regularizations of this type, we investigate the eigenvalues again by examining the solutions of

$$\det \begin{bmatrix} D - (1-\lambda)\bar{D} & \lambda A^T \\ \lambda A & (1-\lambda)D^+ \end{bmatrix} = 0. \tag{4.4}$$

Since $D^+_{ii} > 0$, no eigenvalue is 0 in this case. But it is easy to see that, similar to the previous case, all solutions of (4.4) are real numbers and the same conditions for iterative refinement are valid. In particular, all eigenvalues of R are less than 1, since for $\lambda \geq 1$ the matrix $D - (1-\lambda)\bar{D}$ is positive definite and $(1-\lambda)D^+$ is negative definite. Thus, $(1-\lambda)D^+ - \lambda^2 A \left(D - (1-\lambda)\bar{D}\right)^{-1} A^T$ is negative definite and (4.4) cannot hold for $\lambda \geq 1$.

Furthermore, if $\bar{D}$ is chosen such that

$$\max \frac{D_{ii}}{\bar{D}_{ii}} < 2,$$

then all eigenvalues will be greater than -1, since for $\lambda \leq -1$, the matrix $D - (1-\lambda)\bar{D}$ is negative definite, $(1-\lambda)D^+$ is positive definite, and $(1-\lambda)D^+ - \lambda^2 A \left(D - (1-\lambda)\bar{D}\right)^{-1} A^T$ is positive definite. This shows that dual regularization does not affect the convergence conditions.

In [22] Saunders perturbed the diagonal of M by a small multiples of I in order to permit the use of existing sparse Cholesky factorization software and ensure sufficient stability. The key feature of our approach is that it is more adaptive: the diagonal values of $\bar{D}$ and D^+ can all be modified by a different amount, bounded in order to ensure convergence of the refinement process.

Finally, let us note that a special case of regularization was discussed in [15] for handling free variables in interior point methods. In this case zero diagonal elements in D were replaced by positive ones. Note that condition (3.2) in this case holds for any $\bar{D}_{ii} > 0$ value.

5. Regularization in practice. In our implementation we intend to achieve 10^{-8} relative error in the Euclidean norm when solving system (2.3):

$$\left\| \begin{bmatrix} D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} - \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right\| \leq 10^{-8} \max\left(1, \left\| \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right\|\right).$$

We use a direct factorization approach with iterative refinement, which is stopped if the desired accuracy is achieved or if the Euclidean norm of the residual increases. The strategy for choosing $\bar{D}$ plays an important role in our approach, because it balances stability and efficiency. Increasing the diagonals in $\bar{D}$ may result in more stable factorizations, but may make the iterative refinement process longer. Here we describe an adaptive procedure to determine $\bar{D}$ at each iteration.

Our practical assumption is that improving the condition of $\bar{D}$ reduces the ill-conditioning of $A\bar{D}^{-1}A^T$ and $\bar{M}$. Our approach sets $\bar{D}$ such that its condition number is kept below an adaptively determined threshold with small modification of D. In the case of the factorization of the normal equations system, modification like this reduces the quantity $\text{cond}(A)^2\text{cond}(\bar{D})$, which is an upper bound for the condition number of $A\bar{D}^{-1}A^T$, while for the indefinite Cholesky factorization of (4.3) it reduces the upper bound [22] on the effective condition of $\bar{M}$ [21].

At iteration k, we define the tentative partition P^k of the primal variables that are positive at the optimum, as

$$P^k = \left\{ j \in \{1, \ldots, n\} : \frac{|\Delta x_j^a|}{x_j} \leq \frac{|\Delta z_j^a|}{z_j} \right\}, \tag{5.1}$$

where $(\Delta x^a, \Delta z^a)$ is the primal-dual affine scaling direction at the previous iteration [14]. Furthermore, let $N^k = \{1, \ldots, n\} \setminus P^k$. We chose this indicator because it is independent of problem scaling and has been justified by both theory and practice [1]. Since $A_{N^k} D_{N^k}^{-1} A_{N^k}^T$ vanishes as the IPM approaches the optimal solution, we assume that variables defined by N^k play a less important role, and we concentrate on the behavior of the variables defined by P^k. Thus, we consider the "important" part of the augmented system

$$\begin{pmatrix} D_{P^k} & A_{P^k}^T \\ A_{P^k} & 0 \end{pmatrix},$$

and monitor the quantity $\text{cond}(D_{P^k})$. In our regularization scheme we set $\bar{D}_{N^k} = D_{N^k}$ and regularize D_{P^k} if necessary. We divide the interior point iteration process into two parts. In the first part we monitor the stability during the iterations and determine the largest value for $\text{cond}(D_{P^k})$, referred to as ω, that still results in numerically "safe" computations. There is no regularization applied in this phase. The second phase is called the "regularization phase" and it is activated if system (2.3) cannot be solved to a numerically sufficient accuracy by iterative refinement. In this phase we derive $\bar{D}$ from D and ω and may modify ω if necessary. Our process can be described as follows:

1. Monitoring phase.
   - Set $\omega = 1$ and $k = 0$.
   - At iteration k computes the affine scaling direction (2.3) [13].
     - If iterative refinement converges rapidly, consider the system as numerically "safe" and set $\omega = \max(\omega, \text{cond}(D_{P^k}))$. Continue with the IPM iterations, set $k \longleftarrow k + 1$.

- If iterative refinement converges, but the convergence is rather slow, leave ω unchanged. Continue with the IPM iterations, set $k \longleftarrow k+1$.
- If iterative refinement does not converge, switch to the "regularization phase."

2. Regularization phase.
   - Define $\bar{D}_{N^k} = D_{N^k}$.
   - Set $\delta = \max_{i \in P^k} (D_{ii})$.
   - For all $i \in P^k$ set $\bar{D}_{ii} = \max(D_{ii},\ \delta/\omega)$.
   - Determine γ from (4.2).
   - Compute the factorization of $\bar{M}_\gamma$ and compute the affine scaling direction (2.3).
     - If iterative refinement converges, leave ω unchanged and continue with the IPM iterations, set $k \longleftarrow k+1$.
     - If iterative refinement does not converge, increase ω and restart the computation of iteration k.

It can be observed that δ becomes large during the iterations. Note that increasing ω in the regularization phase may be necessary to compensate for the increase of $cond(A_{P^k})$, relative to $cond(A_{P^{k-1}})$. In our implementation we multiply ω by 10 in a case like this. Also note that the regularization phase may be skipped if sufficient numerical accuracy is achieved for (2.3) during all interior point iterations. This is the case on all NETLIB [5] and QPLIB [11] problems with our solver, for example.

6. Numerical results. We demonstrate the usability of our regularization scheme by solving numerically difficult problems. Let us note that the present NETLIB and QPLIB problem sets do not contain any problems with numerical challenges for modern interior point solvers, and all of these problems are easily solvable without the regularization techniques. We assembled a set of challenging problems from academic and industrial applications. Those available for the public can be accessed from http://www.sztaki.hu/~meszaros/public_ftp/lptestset/. Table 6.1 gives the characteristics of the problems before and after our presolve procedure. During presolve the problem data was scaled by rows and columns as described in [18]. Note that none of these problems was solvable by our implementation BPMPD [16] without the described regularization scheme.

The problem statistics show that numerical difficulties are not related to the problem size, as our test set contains small problems with a few hundred variables, as well as large ones with up to 2 million variables.

Table 6.2 summarizes the numerical properties of the test problems. We denote the partition (5.1) of the optimal solution by P^* and use notation $X = \mathrm{diag}(x_1, \ldots, x_n)$. Thus, the second column of Table 6.2 presents the ratio of the largest and smallest primal values among the primal variables that are positive in the optimal solution. The third column presents the condition number of the scaling matrix D_{P^*} at the last iterate of the interior point method. The fourth column presents the value of ω used for the last iteration. Finally, the iteration numbers of the monitoring and regularization phases are given.

The results indicate that most of the problems are very badly scaled for interior point methods, resulting in badly conditioned scaling matrices D. The results also explain why refinement methods for the normal equation system do not work in these cases: during the computation of $A_{P^*} D_{P^*}^{-1} A_{P^*}^T$ a significant amount of information is lost because of the bad conditioning of D_{P^*}. For some problems, the Cholesky

TABLE 6.1
Problem statistics.

Problem	Original			Preprocessed		
name	rows	columns	nonzeros	rows	columns	nonzeros
11per	19183	28071	207408	8723	15371	156746
1pren	1589	1793	25898	1035	1245	54851
asphalt	5146	4653	40660	1651	1547	89893
check1	216	158	1181	194	156	1177
check2	216	158	1181	194	156	1177
felici203	25301	33865	511720	19492	24508	459678
gmp	86968	460554	1753245	83835	371983	1595884
icplp	2161	66707	555906	1824	9152	51858
l30	2701	15380	51169	1209	13888	698704
mylp	2162	66707	618563	1824	9152	52353
sgpf3y5	30458	39867	103090	17374	26746	61460
ss75	57601	85275	413025	24581	68979	1522662
stat96v1	5997	197472	780606	867	185764	3566191
stat96v2	29091	957432	3783000	4379	917760	18340701
stat96v4	3174	62212	490473	3170	62209	469730
sa4	11101	1990672	21391159	11101	1990672	21391159
sanom	12796	231711	2427152	12368	231662	2420573
vtx202	37921	391594	2721175	31048	384536	6937717

TABLE 6.2
Numerical characteristics of the problems.

Problem	Numerical properties			Iterations	
name	cond(X_{P^*})	cond(D_{P^*})	ω	monitor	regul.
11per	5.6e+14	7.7e+27	2.1e+15	21	57
1pren	3.4e+16	4.6e+32	6.2e+18	14	3
asphalt	4.1e+11	8.7e+22	2.2e+17	32	6
check1	9.1e+07	5.8e+15	1.0e+15	10	17
check2	1.3e+08	2.8e+16	1.0e+15	18	13
felci203	2.0e+10	3.6e+20	1.0e+16	33	13
gmp	1.5e+17	1.5e+29	8.3e+17	44	13
icplp	4.0e+10	7.3e+19	1.2e+17	20	26
l30	4.9e+09	1.5e+20	1.0e+11	12	17
mylp	9.9e+09	5.2e+18	1.0e+14	20	9
sgpf3y5	7.5e+10	9.8e+22	1.0e+11	8	42
ss75	7.8e+13	3.8e+26	1.0e+12	13	46
stat96v1	2.5e+13	2.3e+27	1.0e+11	10	70
stat96v2	5.0e+12	2.6e+24	1.0e+11	6	61
stat96v4	4.6e+10	9.4e+20	1.0e+16	25	5
sa4	9.5e+08	2.8e+16	1.0e+15	36	3
sanom	7.6e+07	1.9e+16	1.0e+15	54	1
vtx202	1.2e+12	1.1e+24	1.6e+17	22	45

factorization is sufficiently accurate up to the last few iterations, such as on problems ,. .96 4, 4, and . On several problems, however, the Cholesky factorization broke down at an early stage of the iteration process, leaving significant work for the regularization phase. Problems like this include 11 ,. 3 5 75 ,. .96 1, and ,. .96 2.

Interior point algorithms terminate when the first-order optimality conditions are satisfied with some predetermined tolerance. This is translated to the following conditions imposed on the relative primal and dual feasibility and the relative duality

TABLE 6.3
Accuracy of the IPM solutions.

Problem name	Relative infeasibility primal	dual	Relative duality gap	Average refinements
11per	1.7e-09	1.3e-12	4.1e-10	28.6
1pren	2.4e-10	5.7e-07	1.0e-14	44.0
asphalt	4.3e-09	3.1e-11	2.3e-09	0.0
check1	8.5e-13	1.0e-12	3.5e-08	4.7
check2	6.6e-13	2.2e-16	3.4e-09	5.5
felci203	4.8e-13	2.1e-16	5.7e-09	12.5
gmp	1.1e-05	5.7e-13	1.0e-06	48.0
icplp	5.9e-11	5.4e-16	1.6e-08	31.3
l30	6.3e-10	5.6e-12	4.7e-09	35.9
mylp	5.1e-08	1.3e-09	1.9e-09	36.0
sgpf3y5	2.5e-15	1.4e-11	3.3e-08	13.7
ss75	2.9e-13	2.4e-09	7.7e-09	36.1
stat96v1	1.0e-11	2.0e-12	3.2e-08	19.1
stat96v2	1.1e-08	2.1e-10	2.2e-07	19.1
stat96v4	3.7e-09	2.1e-17	3.0e-08	2.4
sa4	3.7e-11	8.0e-16	3.0e-09	55.7
sanom	1.1e-09	1.6e-16	8.6e-08	75.9
vtx202	1.5e-10	7.1e-09	1.3e-08	24.1

gap:

$$\frac{||Ax-b||}{1+||b||} \le \epsilon, \quad \frac{||A^T y - c||}{1+||c||} \le \epsilon, \quad \frac{\left|c^T x - b^T y\right|}{1+|b^T y|} \le \epsilon,$$

where ϵ is a positive tolerance. We stopped our IPM implementation if ϵ of order 10^{-8} was achieved, or if the accuracy of the solution was not decreased by at least one order of magnitude during the last 10 iterations. Table 6.3 displays the relative primal and dual infeasibility and the relative duality gap at the termination. Furthermore, the last column of Table 6.3 shows the average number of iterative refinement steps in the regularization phase for computing the "composite" infeasible predictor-corrector direction [9]. The results show that the desired accuracy was achieved on the majority of the problems, while the number of necessary iterative refinement steps was kept moderate. The numerically most ill-conditioned problem appeared to be , but taking into account the condition number of D_{P^*} observed on this problem, the achieved accuracy can be considered to be adequate.

Note that the iterative refinement converged in some cases very rapidly. Interestingly, on problem , no refinement steps were necessary during the regularization phase. Most of the refinement steps were necessary on problems and 4, but these problems required only few iterations in the computationally more expensive regularization phase. The rest of the problems required a moderate number of refinements.

7. Summary. In [17] it was pointed out for interior point methods that if the values of the strictly positive components in the primal optimal solution vector are of different orders of magnitude, then it presents numerically challenging cases for current implementation technology, based on the modified Cholesky decomposition. Here we described a regularization scheme for handling this situation and for solving numerically challenging linear programming problems. We proved that, with some conditions for the regularization, one can compute search directions using a factorization of the regularized system and iterative refinement. A relaxation technique was

described whose application can relax the derived condition and may increase the rate of convergence of the iterative refinement. We discussed several special aspects and properties of the regularization scheme and showed that in some cases the convergence of the iterative refinement can be guaranteed in a finite number of steps because of the nilpotent property of the underlying transformation matrix. Based on an upper bound on the effective condition number of the augmented system $\overline{M}$ (12), we provided an adaptive control mechanism for the regularization. Our scheme classifies the ill-conditioned part of $\overline{M}$ in a scaling-independent way and adaptively determines the necessary amount of regularization.

Our numerical experiments confirmed that the regularization scheme improves the numerical stability and makes interior point implementations more robust.

Acknowledgments. We would like to thank the anonymous referees for their careful reading and valuable advice.

REFERENCES

[1] E. D. Andersen, J. Gondzio, C. Mészáros, and X. Xu, *Implementation of interior point methods for large scale linear programs*, in Interior Point Methods of Mathematical Programming, T. Terlaky, ed., Appl. Math. Optim. 33, Kluwer Acad. Publ., 1996, pp. 189–252.

[2] M. Arioli, I. S. Duff, and P. P. M. de Rijk, *On the augmented system approach to sparse least-squares problems*, Numer. Math., 55 (1989), pp. 667–684.

[3] A. Björk, *Iterative refinement of linear least squares solutions*, BIT, 8 (1967), pp. 8–30.

[4] L. Fox, *An Introduction To Numerical Linear Algebra*, Clarendon Press, Oxford, 1964.

[5] D. M. Gay, *Electronic mail distribution of linear programming test problems*, COAL Newsletter, 13 (1985), pp. 10–12.

[6] P. E. Gill, M. A. Saunders, and J. R. Shinnerl, *On the stability of Cholesky factorization for symmetric quasi-definite systems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 35–46.

[7] J. Gondzio and T. Terlaky, *A computational view of interior point methods for linear programming*, in Advances in Linear and Integer Programming, J. Beasley, ed., Oxford University Press, Oxford, 1995, pp. 103–144.

[8] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point algorithm for linear programming*, in Progress in Mathematical Programming: Interior-Point Algorithms and Related Methods, N. Megiddo, ed., Springer Verlag, New York, 1989, pp. 29–47.

[9] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra Appl., 152 (1991), pp. 191–222.

[10] I. Maros and C. Mészáros, *The role of the augmented system in interior point methods*, Eur. J. Oper. Res., 107 (1998), pp. 720–736.

[11] I. Maros and C. Mészáros, *A repository of convex quadratic programming problems*, Optim. Methods Softw., 11/12 (1999), pp. 671–681. The test problems are available online from ftp://ftp.sztaki.hu/pub/oplab/QPDATA.

[12] N. Megiddo, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming: Interior-Point Algorithms and Related Methods, N. Megiddo, ed., Springer Verlag, New York, 1989, pp. 131–158.

[13] S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM J. Optim., 2 (1992), pp. 575–601.

[14] S. Mehrotra and Y. Ye, *Finding an interior point in the optimal face of linear programs*, Math. Programming, 62 (1993), pp. 497–515.

[15] C. Mészáros, *On free variables in interior point methods*, Optim. Methods Softw., 9 (1997), pp. 121–139.

[16] C. Mészáros, *The BPMPD interior-point solver for convex quadratic problems*, Optim. Methods Softw., 11/12 (1999), pp. 431–449.

[17] C. Mészáros, *On the Cholesky factorization in interior point methods*, Comput. Math. Appl., 50 (2005), pp. 1157–1166.

[18] C. Mészáros and U.H. Suhl, *Advanced preprocessing techniques for linear and quadratic programming*, OR Spectrum, 25 (2004), pp. 575–595.

[19] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and least-squares problems*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[20] M. A. Saunders, *Major Cholesky would feel proud*, ORSA J. Comput., 6 (1994), pp. 94–105.
[21] M. A. Saunders, *Solution of sparse rectangular systems using LSQR and CRAIG*, BIT, 35 (1995), pp. 588–604.
[22] M. A. Saunders, *Cholesky-based methods for sparse least squares: The benefits of regularization*, in Linear and Nonlinear Conjugate Gradient-Related Methods, L. Adams and J. L. Nazareth, eds., SIAM, 1996, pp. 92–100.
[23] R. J. Vanderbei, *Symmetric quasi-definite matrices*, SIAM J. Optim., 5 (1995), pp. 100–113.
[24] M. H. Wright, *Some properties of the Hessian of the logarithmic barrier function*, Math. Programming, 67 (1994), pp. 265–295.
[25] S. J. Wright, *Primal-Dual Interior Point Methods*, SIAM, Philadelphia, 1997.
[26] S. J. Wright, *Stability of augmented system factorizations in interior-point methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 191–222.
[27] S. J. Wright, *Modified Cholesky factorizations in interior-point algorithms for linear programming*, SIAM J. Optim., 9 (1999), pp. 1159–1191.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 236–260

ORTHOGONALIZATION VIA DEFLATION: A MINIMUM NORM APPROACH FOR LOW-RANK APPROXIMATIONS OF A MATRIX*

ACHIYA DAX†

Abstract. In this paper we introduce a new orthogonalization method. Given a real $m \times n$ matrix A, the new method constructs an SVD-type decomposition of the form $A = \hat{U}\hat{\Sigma}\hat{V}^T$. The columns of $\hat{U}$ and $\hat{V}$ are orthonormal, or nearly orthonormal, while $\hat{\Sigma}$ is a diagonal matrix whose diagonal entries approximate the singular values of A. The method has three versions: a "left-side" orthogonalization scheme in which the columns of $\hat{U}$ constitute an orthonormal basis of Range(A), a "right-side" orthogonalization scheme in which the columns of $\hat{V}$ constitute an orthonormal basis of Range(A^T), and a third version in which both $\hat{U}$ and $\hat{V}$ have orthonormal columns, but the decomposition is not exact. The new decompositions may substitute the SVD in many applications.

Key words. orthogonalization via deflation, low-rank approximations, rectangular iterations, missing data estimation

AMS subject classifications. 15A18, 15A23, 65F25, 65F30, 65F50

DOI. 10.1137/060656401

1. Introduction.

In this paper we introduce a new orthogonalization method. Let A be a real $m \times n$ matrix. The new method constructs an SVD-type decomposition of the form

$$A = \hat{U}\hat{\Sigma}\hat{V}^T,$$

where the columns of $\hat{U}$ and $\hat{V}$ are orthonormal (or nearly orthonormal) and $\hat{\Sigma}$ is a diagonal matrix whose diagonal entries approximate the singular values of A. The name of the method, "orthogotheonalization via deflation," comes from the similarity to Hotelling's deflation by subtraction method. Given a symmetric positive semidefinite matrix S, the last method computes the eigenpairs of S, one after another in decreasing order, using the power method to compute dominant eigenpairs of the deflated matrices, e.g., [16], [17], [26], [32]. It is well known, however, that the convergence of the power method can be very slow. This raises the question of whether there are better ways to compute a dominant eigenpair. In the general case we need an effective method for calculating a dominant pair of singular vectors.

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ denote the eigenvalues of S. Then a dominant eigenvector of S satisfies $S\mathbf{q} = \lambda_1\mathbf{q}$ and $\mathbf{q} \neq \mathbf{0}$. The minimum norm approach proposed in this paper computes $\mathbf{q}$ by solving the minimum norm problem

$$\text{minimize } E(\mathbf{q}) = \|S - \mathbf{q}\mathbf{q}^T\|_F^2, \tag{1.1}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm and $\mathbf{q} = (q_1, q_2, \ldots, q_n)^T$ denotes the vector of unknowns. An equivalent way to write (1.1) is

$$\text{minimize } E(q_1, \ldots, q_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} (s_{ij} - q_i q_j)^2, \tag{1.2}$$

*Received by the editors April 5, 2006; accepted for publication (in revised form) by D. Calvetti October 5, 2007; published electronically March 5, 2008.
http://www.siam.org/journals/simax/30-1/65640.html
†Hydrological Service, P.O.B. 36118, Jerusalem 91360, Israel (dax20@water.gov.il).

where s_{ij} denotes the (i,j) entry of S. If $\mathbf{q}_1$ is a dominant eigenvector of S, then the vector $\mathbf{q}^* = (\lambda_1)^{1/2}\mathbf{q}_1/\|\mathbf{q}_1\|_2$ solves (1.1). The converse is also true: Let $\mathbf{q}^*$ solve (1.1), and then $\mathbf{q}^*$ is a dominant eigenvector of S. These observations are easily derived from the symmetric quotients equality that we establish in section 4. Here and henceforth

$$\|\mathbf{q}\|_2 = (\mathbf{q}^T\mathbf{q})^{1/2} = \left(\sum_{j=1}^{n} q_j^2\right)^{1/2}$$

denotes the Euclidean vector norm. Casting the problem of calculating a dominant eigenpair as a minimum norm problem opens the door for effective minimization techniques. For example, minimizing $E(q_1,\ldots,q_n)$ by changing one variable at a time results in a "point relaxation" algorithm that requires about the same computational effort per iteration as the power method but enjoys a faster rate of convergence. See [8] for computational details and numerical experiments.

A similar approach is proposed for calculating a dominant pair of singular vectors of a real $m \times n$ matrix A. In this case we consider the minimum norm problem

$$\text{minimize } F(\mathbf{u},\mathbf{v}) = \|A - \mathbf{u}\mathbf{v}^T\|_F^2, \tag{1.3}$$

where $\mathbf{u} = (u_1, u_2, \ldots, u_m)^T \in \mathbb{R}^m$ and $\mathbf{v} = (v_1, v_2, \ldots, v_n)^T \in \mathbb{R}^n$ denote the vectors of unknowns. An alternative way to write (1.3) is

$$\text{minimize } F(u_1,\ldots,u_m,v_1,\ldots,v_n) = \sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - u_i v_j)^2, \tag{1.4}$$

where a_{ij} denotes the (i,j) entry of A. To simplify our notations we make the assumption that $m \geq n$. In this case A has an SVD of the form

$$A = U\Sigma V^T, \tag{1.5}$$

where $U = [\mathbf{u}_1,\ldots,\mathbf{u}_m]$ is an $m \times m$ orthogonal matrix, $V = [\mathbf{v}_1,\ldots,\mathbf{v}_n]$ is an $n \times n$ orthogonal matrix, and $\Sigma = \text{diag}\{\sigma_1,\ldots,\sigma_n\}$ is an $m \times n$ diagonal matrix. There is no loss of generality in assuming that the singular values are nonnegative and sorted to satisfy

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0.$$

The columns of U are called left singular vectors, while the columns of V are called right singular vectors. A pair of nonzero vectors that satisfy

$$\|\hat{\mathbf{u}}\|_2 = 1, \quad \|\hat{\mathbf{v}}\|_2 = 1, \quad A\hat{\mathbf{v}} = \sigma_1\hat{\mathbf{u}}, \quad \text{and} \quad A^T\hat{\mathbf{u}} = \sigma_1\hat{\mathbf{v}}$$

is called a dominant singular pair. In this case the vectors $\mathbf{u}^* = (\sigma_1)^{1/2}\hat{\mathbf{u}}$ and $\mathbf{v}^* = (\sigma_1)^{1/2}\hat{\mathbf{v}}$ solve (1.3). The converse claim is also true: If the vectors $\mathbf{u}^*$ and $\mathbf{v}^*$ solve (1.3), then $\hat{\mathbf{u}} = \mathbf{u}^*/\|\mathbf{u}^*\|_2$ and $\hat{\mathbf{v}} = \mathbf{v}^*/\|\mathbf{v}^*\|_2$ form a dominant singular pair with $\sigma_1 = \hat{\mathbf{u}}^T A\hat{\mathbf{v}}$. These observations are a corollary of the well-known Eckart–Young theorem, e.g., [3] or [11]. (The last theorem is also called the Schmidt–Mirsky theorem, e.g., [28].) The rectangular quotient equality that we establish in section 5 provides an alternative way to conclude these observations.

Recall that (1.5) implies the equality

$$A = \sum_{j=1}^{n} \sigma_j \mathbf{u}_j \mathbf{v}_j^T, \tag{1.6}$$

while a partial sum of the form

$$B_\ell = \sum_{j=1}^{\ell} \sigma_j \mathbf{u}_j \mathbf{v}_j^T \tag{1.7}$$

is called a low-rank approximation of order ℓ (also called truncated SVD). The orthogonal decomposition proposed in this paper has similar structure and may replace the SVD in many applications. It is especially attractive in cases when standard SVD algorithms are not applicable.

The paper is divided into two parts. The first one concentrates on symmetric positive semidefinite matrices. The second part extends the results to general $m \times n$ matrices. This helps to see the motivation behind the proposed methods and the close links between the two cases. The plan of the paper is as follows. It starts with a brief overview of the power method and its basic features. By using deflation by subtraction the power method is harnessed to yield a complete eigensystem of S. The modified deflation scheme proposed in section 3 is essentially an orthogonalization process that has important advantages over the classical deflation by subtraction process.

The second part of the paper starts by extending the Rayleigh quotient to real $m \times n$ matrices. Given two vectors $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^n$, the [illegible]

$$\rho(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T A \mathbf{v} / (\|\mathbf{u}\|_2 \|\mathbf{v}\|_2) \tag{1.8}$$

estimates the corresponding singular value of A. The [illegible], which is proved in section 5, connects the minimum norm problem (1.3) with $\rho(\mathbf{u}, \mathbf{v})$, showing that minimizing $\|A - \mathbf{u}\mathbf{v}^T\|_F$ is equivalent to maximizing $\rho(\mathbf{u}, \mathbf{v})$. The [illegible] that we propose solve this problem in an effective way. The new orthogonalization process is introduced in section 8. An advantage of the proposed approach is its ability to handle problems in which some entries of A are missing. This issue is briefly discussed in section 9.

2. The power method and deflation by subtraction. As before, S denotes a symmetric positive semidefinite matrix of order n with eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0.$$

Let $\mathbf{q}_1, \mathbf{q}_2, \ldots \mathbf{q}_n$ denote the corresponding system of orthonormal eigenvectors. That is, $\|\mathbf{q}_j\|_2 = 1$ and $S\mathbf{q}_j = \lambda_j \mathbf{q}_j$ for $j = 1, \ldots, n$, and $\mathbf{q}_i^T \mathbf{q}_j = 0$ when $i \neq j$. The power method is an iterative algorithm for computing a dominant eigenpair of S. The deflation by subtraction process enables us to harness the power method for calculating the other eigenpairs of S. In this section we briefly overview the basic features of these methods.

2.1. The power method. Starting with some initial unit vector $\mathbf{p}_0$, the kth iteration $k = 1, 2, \ldots$ is composed of the following three steps:

1. Compute $\mathbf{w}_k = S\mathbf{p}_{k-1}$.
2. Compute $\rho_k = \mathbf{p}_{k-1}^T \mathbf{w}_k$.
3. Compute $\mathbf{p}_k = \mathbf{w}_k / \|\mathbf{w}_k\|_2$.

The definition of $\mathbf{p}_k$ implies the equality

$$\mathbf{p}_k = S^k\mathbf{p}_0/\|S^k\mathbf{p}_0\|_2,$$

while the starting vector has a unique presentation in the form

$$\mathbf{p}_0 = \alpha_1\mathbf{q}_1 + \alpha_2\mathbf{q}_2 + \cdots + \alpha_n\mathbf{q}_n,$$

where $\alpha_1, \ldots, \alpha_n$ are real numbers. So

$$S^k\mathbf{p}_0 = \alpha_1\lambda_1^k\mathbf{q}_1 + \alpha_2\lambda_2^k\mathbf{q}_2 + \cdots + \alpha_n\lambda_n^k\mathbf{q}_n.$$

Thus, when $\alpha_1 \neq 0$ and $\lambda_1 > \lambda_2$, the sequence $\{\mathbf{p}_k\}$ converges toward $\mathbf{q}_1$ at a linear rate proportional to the ratio λ_2/λ_1. The definition of ρ_k implies the relations

$$\begin{aligned}\rho_{k+1} &= \mathbf{p}_k^T S\mathbf{p}_k = (S^k\mathbf{p}_0)^T S(S^k\mathbf{p}_0)/(S^k\mathbf{p}_0)^T S^k\mathbf{p}_0 \\ &= \left(\sum_{j=1}^n \alpha_j^2\lambda_j^{2k+1}\right)\Bigg/\left(\sum_{j=1}^n \alpha_j^2\lambda_j^{2k}\right) \\ &= \lambda_1\left(\sum_{j=1}^n \alpha_j^2\beta_j^{2k+1}\right)\Bigg/\left(\sum_{j=1}^n \alpha_j^2\beta_j^{2k}\right),\end{aligned}$$

where $\beta_j = \lambda_j/\lambda_1$, $j = 1, \ldots, n$. The last equality shows that

$$0 \leq \rho_k \leq \lambda_1 \quad \text{for } k = 1, 2, \ldots .$$

It is also easy to verify that

$$\lambda_1 - \rho_{k+1} \leq (\lambda_2/\lambda_1)^2(\lambda_1 - \rho_k) \tag{2.1}$$

and

$$0 \leq \rho_1 \leq \rho_2 \leq \cdots \leq \rho_k \leq \rho_{k+1} \leq \cdots \leq \lambda_1. \tag{2.2}$$

So the sequence $\{\rho_k\}$ converges monotonously toward λ_1 at a linear rate proportional to $(\lambda_2/\lambda_1)^2$.

Of course, if λ_j is considerably smaller than λ_2, then the jth component of $\mathbf{p}_k$ vanishes at a much faster rate. Therefore, when a large portion of $\mathbf{p}_0$ is related to "small" eigenvalues, the initial rate of convergence is much faster than the asymptotic rate, which depends on the ratio λ_2/λ_1. More precisely, assume for a moment that α_1 is not too small and that λ_j is a small eigenvalue that satisfies $\lambda_j/\lambda_1 \leq 1/2$. In this case the size of the product $\mathbf{q}_j^T\mathbf{p}_k$ is, at least, halved every iteration. Thus, unless $|\mathbf{q}_j^T\mathbf{p}_0|$ is much larger than $|\mathbf{q}_1^T\mathbf{p}_0|$, the product $\mathbf{q}_j^T\mathbf{p}_k$ actually vanishes after a small number of iterations. In other words, unless $\mathbf{p}_0$ is nearly perpendicular to $\mathbf{q}_1$, a few power iterations are likely to wipe out components of small eigenvectors, turning $\mathbf{p}_k$ into a linear combination of "large" eigenvectors. Consequently ρ_k provides a good estimate of λ_1 within a small number of iterations. This feature is illustrated in Table 2.1 for various distributions of eigenvalues. The test problems are defined with $\lambda_1 = 1$ and $\alpha_j = 1/\sqrt{n}$, $j = 1, \ldots, n$. The figures in Table 2.1 give the values of $\rho_5, \rho_{10}, \rho_{20}$, and ρ_{40}, rounded to four decimals. Thus, for example, the second row of Table 2.1 considers the case when $n = 200$ and $\lambda_j = (201 - j)/200$, $j = 1, \ldots, 200$. In this

TABLE 2.1
Initial rates of convergence of the power method.

Matrix $\lambda_1 = 1$, $\lambda_j,\ j = 2, \ldots, n$	n	Values of ρ_k $k = 5$	$k = 10$	$k = 20$	$k = 40$
$\lambda_j = (n+1-j)/n$	20	0.9375	0.9744	0.9929	0.9922
$\lambda_j = (n+1-j)/n$	200	0.9189	0.9569	0.9786	0.9901
$\lambda_j = (1/2)^{j-1}$	20	0.9995	>0.9999	>0.9999	>0.9999
$\lambda_j = (1/2)^{j-1}$	200	0.9995	>0.9999	>0.9999	>0.9999
$\lambda_j = (1/\sqrt{2})^{j-1}$	20	0.9906	0.9997	>0.9999	>0.9999
$\lambda_j = (1/\sqrt{2})^{j-1}$	200	0.9906	0.9997	>0.9999	>0.9999
$\lambda_j = (0.9)^{j-1}$	20	0.9492	0.9863	0.9985	>0.9999
$\lambda_j = (0.9)^{j-1}$	200	0.9492	0.9863	0.9985	>0.9999
$\lambda_j = (0.99)^{j-1}$	20	0.9393	0.9602	0.9802	0.9920
$\lambda_j = (0.99)^{j-1}$	200	0.9136	0.9570	0.9802	0.9920
$\lambda_j = (0.999)^{j-1}$	20	0.9909	0.9912	0.9919	0.9931
$\lambda_j = (0.999)^{j-1}$	200	0.9353	0.9561	0.9762	0.9881
Random λ_j	20	0.9528	0.9722	0.9829	0.9934
Random λ_j	200	0.9256	0.9630	0.9814	0.9903

case the ratio $\lambda_2/\lambda_1 = 0.995$ dictates a slow asymptotic rate of convergence. Yet the resulting values of ρ_k are $\rho_5 = 0.9189,\ \rho_{10} = 0.9569,\ \rho_{20} = 0.9786$, and $\rho_{40} = 0.9901$. That is, ρ_k provides a fair estimate of $\lambda_1 = 1$ after a small number of iterations, in spite of slow asymptotic convergence. For a detailed discussion of this feature see [21].

Although the power method is well known, the question of how to choose the starting point $\mathbf{p}_0$ has no definite answer, especially when no approximation to $\mathbf{q}_1$ is available. In such a case taking $\mathbf{p}_0$ to be a random vector is a reasonable option, e.g., [10], [27], or [29]. The stopping condition can be based on the error bound (4.7).

2.2. Deflating by subtraction. This method is based on the following idea. Consider the spectral decomposition

$$S = \sum_{j=1}^{n} \lambda_j \mathbf{q}_j \mathbf{q}_j^T, \tag{2.3}$$

and let the matrices $S_1, S_2, \ldots, S_n$, be defined by the rule: $S_1 = S$ and

$$S_{\ell+1} = S_\ell - \lambda_\ell \mathbf{q}_\ell \mathbf{q}_\ell^T, \qquad \ell = 1, \ldots, n-1.$$

Then $S_\ell,\ \ell = 1, \ldots, n$, is a symmetric positive semidefinite matrix whose largest eigenvalue is λ_ℓ. The power method enables us to compute a dominant eigenpair of S_ℓ and to construct $S_{\ell+1}$. This way the eigenpairs of S are computed one after another in decreasing order. The practical implementation of this idea is carried out as follows. The deflation process is composed of (at most) n stages. The ℓth stage, $\ell = 1, \ldots, n$, starts with a matrix $\tilde{S}_\ell$ and ends with $\tilde{S}_{\ell+1}$. (The matrix $\tilde{S}_\ell$ denotes a computed estimate of S_ℓ.) At the ℓth deflation stage the power method is applied to the matrix $\tilde{S}_\ell$ to provide an estimated dominant eigenpair of $\tilde{S}_\ell$. (The starting point and the number of iterations can be arbitrary.) Let $\tilde{\lambda}_\ell$ and $\tilde{\mathbf{q}}_\ell,\ \|\tilde{\mathbf{q}}_\ell\|_2 = 1$, denote the computed eigenpair. Then $\tilde{S}_{\ell+1}$ is constructed by the rule

$$\tilde{S}_{\ell+1} = \tilde{S}_\ell - \tilde{\lambda}_\ell \tilde{\mathbf{q}}_\ell \tilde{\mathbf{q}}_\ell^T. \tag{2.4}$$

The idea of deflation by subtraction is often attributed to Hotelling [16], [17]. For further discussions of this method see [26] and [32]. Parlett [26, p. 82] gives a detailed error analysis that quantifies the change in distant eigenvalues caused by the error in the computed dominant eigenpair.

3. Orthogonalization via deflation: The symmetric case. By starting with $\hat{S}_1 = S$, the new deflation by subtraction process constructs a sequence of symmetric matrices, $\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_\ell, \hat{S}_{\ell+1} \ldots$, where $\hat{S}_{\ell+1}$ is obtained from $\hat{S}_\ell$ in the following way. If $\hat{S}_\ell = 0$, the algorithm terminates. Otherwise, we choose a unit vector $\hat{\mathbf{q}}_\ell$ such that

$$\|\hat{\mathbf{q}}_\ell\|_2 = 1 \qquad \text{and} \qquad \hat{\mathbf{q}}_\ell \in \text{Range}(\hat{S}_\ell) \tag{3.1}$$

and define

$$\hat{S}_{\ell+1} = \hat{S}_\ell - (\hat{S}_\ell \hat{\mathbf{q}}_\ell)(\hat{S}_\ell \hat{\mathbf{q}}_\ell)^T / \hat{\mathbf{q}}_\ell^T \hat{S}_\ell \hat{\mathbf{q}}_\ell. \tag{3.2}$$

An alternative way to write (3.2) is

$$\hat{S}_{\ell+1} = \hat{S}_\ell - \mu_\ell \mathbf{u}_\ell \mathbf{u}_\ell^T, \tag{3.3}$$

where

$$\mathbf{u}_\ell = \hat{S}_\ell \hat{\mathbf{q}}_\ell / \|\hat{S}_\ell \hat{\mathbf{q}}_\ell\|_2 \quad \text{and} \quad \mu_\ell = \hat{\mathbf{q}}_\ell^T \hat{S}_\ell^2 \hat{\mathbf{q}}_\ell / \hat{\mathbf{q}}_\ell^T \hat{S}_\ell \hat{\mathbf{q}}_\ell. \tag{3.4}$$

THEOREM 1. [illegible] $\hat{Q}_\ell = [\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_\ell]$ [illegible] $n \times \ell$ [illegible] $\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_\ell$ [illegible] r [illegible] S [illegible] $\ell = 1, 2, \ldots, r$

$$\hat{Q}_\ell^T \hat{Q}_\ell = I, \quad \hat{S}_{\ell+1} \hat{Q}_\ell = 0, \quad \text{and} \quad \text{Range}(\hat{Q}_\ell) \subseteq \text{Range}(S). \tag{3.5}$$

[illegible] $\ell = r$ [illegible] $\hat{Q}_r$ [illegible] Range(S),

$$\hat{S}_{r+1} = 0 \tag{3.6}$$

[illegible]

$$S = \sum_{\ell=1}^{r} \mu_\ell \mathbf{u}_\ell \mathbf{u}_\ell^T = U_r D_r U_r^T, \tag{3.7}$$

[illegible]

$$U_r = [\mathbf{u}_1, \ldots, \mathbf{u}_r] \quad \text{and} \quad D_r = \text{diag}\{\mu_1, \ldots, \mu_r\}.$$

[illegible]. The proof of (3.5) is by induction on ℓ. For $\ell = 1$ the relations in (3.5) follow from the definitions of $\hat{\mathbf{q}}_1$ and $\hat{S}_2$. Assume now that (3.5) holds for $\ell - 1$. That is, $\hat{Q}_{\ell-1}^T \hat{Q}_{\ell-1} = I$, $\hat{S}_\ell \hat{Q}_{\ell-1} = 0$, and Range$(\hat{Q}_{\ell-1}) \subseteq$ Range(S). Assume further that $\hat{\mathbf{q}}_\ell$ and $\hat{S}_{\ell+1}$ satisfy (3.1) and (3.2). Then, clearly, $\hat{S}_{\ell+1} \hat{\mathbf{q}}_\ell = \mathbf{0}$ and $\hat{Q}_\ell^T S_{\ell+1} = 0$. Similarly, from $\hat{Q}_{\ell-1} \hat{S}_\ell = 0$ and (3.1) we conclude that $\hat{Q}_\ell^T \hat{Q}_\ell = I$. Finally, since $\hat{\mathbf{q}}_\ell \in$ Range(S) and Range$(\hat{Q}_{\ell-1}) \subseteq$ Range(S), it follows that Range$(\hat{Q}_\ell) \subseteq$ Range(S). □

At the end of the ℓth deflation step we are able to construct two low-rank approximations of S:

$$B_\ell = U_\ell D_\ell U_\ell^T = \sum_{j=1}^{\ell} \mu_j \mathbf{u}_j \mathbf{u}_j^T \tag{3.8}$$

and

$$\hat{B}_\ell = \hat{Q}_\ell \hat{D}_\ell \hat{Q}_\ell^T = \sum_{j=1}^{\ell} \hat{\rho}_j \hat{\mathbf{q}}_j \hat{\mathbf{q}}_j^T, \tag{3.9}$$

where

$$\hat{D}_\ell = \mathrm{diag}\{\hat{\rho}_1, \ldots, \hat{\rho}_\ell\}$$

and $\hat{\rho}_j$ denotes the Rayleigh quotient corresponding to $\hat{\mathbf{q}}_j$. That is,

$$\hat{\rho}_j = \hat{\mathbf{q}}_j S \hat{\mathbf{q}}_j.$$

It is interesting to compare these approximations and the corresponding Hotelling low-rank approximation

$$\tilde{B}_\ell = \tilde{Q}_\ell \tilde{D}_\ell \tilde{Q}_\ell^T = \sum_{j=1}^{\ell} \tilde{\lambda}_j \, \tilde{\mathbf{q}}_j \, \tilde{\mathbf{q}}_j^T. \tag{3.10}$$

Our experience shows that, if both schemes use the same number of power iterations at each deflation step, the differences between $\|S - \tilde{B}_\ell\|_F$, $\|S - B_\ell\|_F$, and $\|S - \hat{B}_\ell\|_F$ remain negligible until $\ell = r$. Yet the new deflation process enjoys important advantages. First, the "finite termination" property (3.6) and the "exactness" property (3.7) provide an effective "rank-revealing" decomposition. Second, the orthogonality of $\hat{Q}_\ell$ and the resulting orthogonal decomposition (3.9) are useful in many applications. Perhaps the more striking feature of the new deflation process is that these properties hold regardless of the quality of $\hat{\mathbf{q}}_\ell$ as a substitute for a dominant eigenvector of $\hat{S}_\ell$. Thus, for example, when $\hat{\mathbf{q}}_\ell$ is computed by applying the power method to $\hat{S}_\ell$, the number of iterations does not effect these properties. Moreover, as we have seen, a few power iterations per eigenpair are sufficient to produce a meaningful estimate of the spectral decomposition. Indeed, preliminary experiments that we have done support this view; see [8].

Let us turn now to see how rounding errors affect the new deflation process. Assume for simplicity that $\mathbf{u}_\ell$ and μ_ℓ provide a fair estimate of a dominant eigenpair of $\hat{S}_\ell$, $\ell = 1, \ldots, r$. Then, on one hand, the size of $\|\hat{S}_{\ell+1}\|_F$ is expected to be about $\lambda_{\ell+1} + \cdots + \lambda_n$. On the other hand, in floating point arithmetic the ℓth deflation step (3.3) adds rounding errors into the entries of $\hat{S}_{\ell+1}$. The size of the resulting error matrix is about the size of the matrix $\varepsilon \mu_\ell \mathbf{u}_\ell \mathbf{u}_\ell^T$, where ε denotes the machine precision (or unit roundoff) in our computations. Hence the overall rounding errors in the entries of $\hat{S}_{\ell+1}$ constitute an error matrix $E_{\ell+1}$, whose Frobenius norm is about $\varepsilon(\lambda_1 + \cdots + \lambda_\ell)$. That is,

$$\|E_{\ell+1}\|_F / \|\hat{S}_{\ell+1}\|_F \approx \varepsilon(\lambda_1 + \cdots + \lambda_\ell)/(\lambda_{\ell+1} + \cdots + \lambda_n). \tag{3.11}$$

This exposes a certain deficiency of the deflation by subtraction approach. If S is an ill-conditioned matrix, then small eigenvalues are computed with a large relative error. The smaller the ratio $\lambda_{\ell+1}/(\lambda_1 + \cdots + \lambda_\ell)$, the larger the relative error in the computed

value of $\lambda_{\ell+1}$. A similar remark applies to the relative error in a computed dominant eigenvector of $\hat{S}_{\ell+1}$.

Another feature that characterizes ill-conditioned matrices is the gradual loss of orthogonality in the columns of $\hat{Q}_\ell$. Observe that every column of the rounding errors matrix $E_{\ell+1}$ is expected to have a component in Range($\hat{Q}_\ell$) whose size is not much smaller than the column's size. Thus, as the ratio (3.11) becomes meaningful, the columns of $\hat{S}_{\ell+1}$ lose their orthogonality against the columns of $\hat{Q}_\ell$. Hence the fact that $\hat{\mathbf{q}}_{\ell+1}$ belongs to Range($\hat{S}_{\ell+1}$) is not sufficient to ensure its orthogonality against the columns of $\hat{Q}_\ell$.

This loss of orthogonality is easily recovered by replacing (3.1) with the following modification. Let $\mathbf{z}_1$ be a unit vector that belongs to Range($\hat{S}_\ell$). (Here $\mathbf{z}_1$ denotes our estimate to a dominant eigenvector of $\hat{S}_\ell$.) Then $\mathbf{z}_1$ is orthogonalized against $\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_{\ell-1}$, by using Gram–Schmidt orthogonalization. This way, one generates a sequence of ℓ vectors, $\mathbf{z}_1, \ldots, \mathbf{z}_\ell$, where $\mathbf{z}_{i+1}$ is obtained from $\mathbf{z}_i$ by the rule

$$\mathbf{z}_{i+1} = \mathbf{z}_i - (\mathbf{z}_i^T \hat{\mathbf{q}}_i)\hat{\mathbf{q}}_i, \quad i = 1, \ldots, \ell - 1. \tag{3.12}$$

In practice $\mathbf{z}_{i+1}$ is overwritten on $\mathbf{z}_i$, so only one vector storage is needed for this process. Once the orthogonalization process is finished, the vector $\hat{\mathbf{q}}_\ell$ which is used in (3.1) is redefined as

$$\hat{\mathbf{q}}_\ell = \mathbf{z}_\ell / \|\mathbf{z}_\ell\|_2. \tag{3.13}$$

(If $\|\mathbf{z}_\ell\|_2$ is smaller than $\|\mathbf{z}_1\|_2/2$, say, then $\mathbf{z}_\ell$ is reorthogonalized against the columns of $\hat{Q}_{\ell-1}$. However, as we have seen, a small number of power iterations is likely to provide a fair estimate of a dominant vector of $\hat{S}_\ell$. In this case $\|\mathbf{z}_\ell\|_2$ is not expected to be much smaller than $\|\mathbf{z}_1\|_2$, so reorthogonalization is seldom needed.)

Summarizing the above discussion we stress that rounding errors cause only tiny perturbations to the eigenvalues of $\hat{S}_{\ell+1}$, as every eigenvalue of $\hat{S}_{\ell+1}$ lies in at least one of the circular discs

$$\{\lambda \,|\, |\hat{\lambda}_i - \lambda| \le \|E_{\ell+1}\|_2\}, \qquad i = 1, \ldots, n, \tag{3.14}$$

where $\hat{\lambda}_1, \ldots, \hat{\lambda}_n$ denote the eigenvalues of $\hat{S}_{\ell+1} - E_{\ell+1}$ and

$$\|E_{\ell+1}\|_2 \le \|E_{\ell+1}\|_F \approx \varepsilon(\lambda_1 + \cdots + \lambda_\ell).$$

Note also that $\|S - U_r D_r U_r^T\|_F$ is not expected to be much larger than $\varepsilon(\lambda_1 + \cdots + \lambda_r)$. Hence the resulting decompositions (3.8) and (3.9) provide valuable information on S, unless one is interested in tiny eigenvalues and their eigenvectors.

4. Symmetric quotients. In this section we introduce a new quantity (the symmetric quotient) and derive its basic features. Let $\mathbf{u} \neq \mathbf{0}$ be a given vector in $\mathbb{R}^n$, and consider the one-parameter function

$$f(\theta) = \|S - \theta \mathbf{u}\mathbf{u}^T\|_F^2. \tag{4.1}$$

Then the [illegible]

$$\gamma = \gamma(\mathbf{u}) = \mathbf{u}^T S \mathbf{u} / (\mathbf{u}^T \mathbf{u})^2 \tag{4.2}$$

provides the value of θ which minimizes $f(\theta)$. That is,

$$\gamma = \arg\min f(\theta). \tag{4.3}$$

The validity of (4.3) is easily verified by rewriting $f(\theta)$ in the form

$$f(\theta) = \|\theta\mathbf{y} - \mathbf{z}\|_2^2,$$

where $\mathbf{y}$ and $\mathbf{z}$ are n^2 vectors. This presentation implies that

$$\arg\min f(\theta) = \mathbf{y}^T\mathbf{z}/\mathbf{y}^T\mathbf{y}.$$

Hence the equalities

$$\mathbf{y}^T\mathbf{z} = \sum_{i=1}^{n}\sum_{j=1}^{n} u_i u_j s_{ij} = \mathbf{u}^T S\mathbf{u}$$

and

$$\mathbf{y}^T\mathbf{y} = \sum_{i=1}^{n}\sum_{j=1}^{n}(u_i u_j)^2 = \|\mathbf{u}\mathbf{u}^T\|_F^2 = (\mathbf{u}^T\mathbf{u})^2$$

prove (4.3).

At this point it is instructive to compare the symmetric quotient with the corresponding [illegible]

$$\rho = \rho(\mathbf{u}) = \mathbf{u}^T S\mathbf{u}/\mathbf{u}^T\mathbf{u}. \tag{4.4}$$

Recall that

$$\rho = \arg\min g(\theta), \tag{4.5}$$

where

$$g(\theta) = \|S\mathbf{u} - \theta\mathbf{u}\|_2^2. \tag{4.6}$$

Another useful feature of $\rho(\mathbf{u})$ is the existence of an eigenvalue λ of S that satisfies the bound

$$|\lambda - \rho(\mathbf{u})| \le \|S\mathbf{u} - \rho(\mathbf{u})\mathbf{u}\|_2/\|\mathbf{u}\|_2; \tag{4.7}$$

see [26, p. 69]. Thus, roughly speaking, $\rho(\mathbf{u})$ approximates the eigenvalue corresponding to $\mathbf{u}$. For a detailed discussion of the Rayleigh quotient and its properties, see [10], [25], [26], [29], [32].

Scaling of $\mathbf{u}$ affects $\rho(\mathbf{u})$ and $\gamma(\mathbf{u})$ in different ways. Let $\alpha \neq 0$ be a given scalar. Then

$$\rho(\alpha\mathbf{u}) = \rho(\mathbf{u}),$$

but

$$\gamma(\alpha\mathbf{u}) = \gamma(\mathbf{u})/\alpha^2.$$

Moreover, define $\mathbf{v} = \alpha\mathbf{u}$, and then

$$\|S - \gamma(\mathbf{v})\mathbf{v}\mathbf{v}^T\|_F^2 = \|S - \gamma(\mathbf{u})\mathbf{u}\mathbf{u}^T\|_F^2.$$

The basic relation that connects $\gamma(\mathbf{u})$ and $\rho(\mathbf{u})$ is the [illegible]

$$\|S - \gamma(\mathbf{u})\mathbf{u}\mathbf{u}^T\|_F^2 = \|S\|_F^2 - (\rho(\mathbf{u}))^2. \tag{4.8}$$

As we have seen, scaling of $\mathbf{u}$ does not affect this equality. Hence, when proving (4.8) there is no loss of generality in assuming that $\|\mathbf{u}\|_2 = 1$. In this case $\gamma = \rho = \mathbf{u}^T S\mathbf{u}$, $\|\mathbf{u}\mathbf{u}^T\|_F = 1$, and

$$\begin{aligned}
\|S - \gamma\mathbf{u}\mathbf{u}^T\|_F^2 &= \|S - \rho\mathbf{u}\mathbf{u}^T\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}(s_{ij} - \rho u_i u_j)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\left[s_{ij}^2 - 2\rho s_{ij}u_i u_j + \rho^2(u_i u_j)^2\right] \\
&= \|S\|_F^2 - 2\rho\mathbf{u}^T S\mathbf{u} + \rho^2\|\mathbf{u}\mathbf{u}^T\|_F^2 \\
&= \|S\|_F^2 - 2\rho^2 + \rho^2 = \|S\|_F^2 - \rho^2.
\end{aligned}$$

The meaning of the symmetric quotient equality (4.8) is that solving the minimum norm problem (1.1) is equivalent to solving the problem

$$\text{maximize } \rho(\mathbf{u}) = \mathbf{u}^T S\mathbf{u}/\mathbf{u}^T\mathbf{u}. \tag{4.9}$$

By using the spectral decomposition (2.3) one can show that a point $\mathbf{u}^* \in \mathbb{R}^n$ solves (4.9) if and only if $\mathbf{u}^*$ is a dominant eigenvector of S, so the optimal value of $\rho(\mathbf{u})$ is $\rho(\mathbf{u}^*) = \lambda_1$. The proof of this claim is outlined at the end of the next section, in which we establish similar results for rectangular matrices. Further extensions of the symmetric quotients equality are derived in [9].

5. Rectangular quotients. In this section we consider the rectangular minimum norm problem (1.3), by using the notations of section 1. We shall start by introducing a new useful quantity, the [illegible]. Given two nonzero vectors $\mathbf{u} = (u_1, \ldots, u_m)^T \in \mathbb{R}^m$ and $\mathbf{v} = (v_1, \ldots, v_n)^T \in \mathbb{R}^n$, the rectangular quotient

$$\eta = \eta(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T A\mathbf{v}/(\|\mathbf{u}\|_2^2\ \|\mathbf{v}\|_2^2) \tag{5.1}$$

solves the one-parameter minimization problem

$$\text{minimize } f(\theta) = \|A - \theta\mathbf{u}\mathbf{v}^T\|_F^2. \tag{5.2}$$

That is,

$$\eta = \arg\min f(\theta). \tag{5.3}$$

To prove the last statement we rewrite (5.2) in the form

$$\text{minimize } f(\theta) = \|\mathbf{a} - \theta\mathbf{w}\|_2^2, \tag{5.4}$$

where here $\mathbf{a}$ and $\mathbf{w}$ are mn-vectors. The last problem has a unique solution at the point

$$\theta^* = \mathbf{w}^T\mathbf{a}/\mathbf{w}^T\mathbf{w}.$$

Hence the equalities

$$\mathbf{w}^T\mathbf{a} = \sum_{i=1}^{m}\sum_{j=1}^{n} u_i v_j a_{ij} = \mathbf{u}^T A\mathbf{v}$$

and

$$\mathbf{w}^T\mathbf{w} = \sum_{i=1}^{m}\sum_{j=1}^{n}(u_i v_j)^2 = \|\mathbf{u}\mathbf{v}^T\|_F^2 = \|\mathbf{u}\|_2^2\,\|\mathbf{v}\|_2^2$$

prove (5.3).

Observe that the rectangular quotient $\eta(\mathbf{u},\mathbf{v})$ does not necessarily provide an estimate of the singular value that "corresponds" to $\mathbf{u}$ and $\mathbf{v}$. For that purpose we introduce another quantity, the [illegible]. Let the unit vectors $\hat{\mathbf{u}} = \mathbf{u}/\|\mathbf{u}\|_2$ and $\hat{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|_2$ be obtained from $\mathbf{u}$ and $\mathbf{v}$, respectively. Then the following three problems:

$$\text{minimize} \quad f_1(\theta) = \|A - \theta\hat{\mathbf{u}}\hat{\mathbf{v}}^T\|_F^2, \tag{5.5}$$

$$\text{minimize} \quad f_2(\theta) = \|A\hat{\mathbf{v}} - \theta\hat{\mathbf{u}}\|_2^2, \tag{5.6}$$

$$\text{minimize} \quad f_3(\theta) = \|A^T\hat{\mathbf{u}} - \theta\hat{\mathbf{v}}\|_2^2, \tag{5.7}$$

share the same solution

$$\hat{\theta} = \hat{\mathbf{u}}^T A\hat{\mathbf{v}} = \mathbf{u}^T A\mathbf{v}/(\|\mathbf{u}\|_2\,\|\mathbf{v}\|_2). \tag{5.8}$$

This observation suggests that the [illegible]

$$\rho = \rho(\mathbf{u},\mathbf{v}) = \mathbf{u}^T A\mathbf{v}/(\|\mathbf{u}\|_2\,\|\mathbf{v}\|_2) \tag{5.9}$$

approximates the singular value of A that corresponds to $\mathbf{u}$ and $\mathbf{v}$.

Let us turn now to see how scaling affects rectangular quotients. Let $\alpha > 0$ and $\beta > 0$ be two positive real numbers. Then

$$\begin{aligned}\rho(\alpha\mathbf{u},\beta\mathbf{v}) &= \rho(\mathbf{u},\mathbf{v}),\\ \eta(\alpha\mathbf{u},\beta\mathbf{v}) &= \eta(\mathbf{u},\mathbf{v})/(\alpha\beta),\end{aligned}$$

and

$$\|A - \eta(\alpha\mathbf{u},\beta\mathbf{v})(\alpha\mathbf{u})(\beta\mathbf{v}^T)\|_F^2 = \|A - \eta(\mathbf{u},\mathbf{v})\mathbf{u}\mathbf{v}^T\|_F^2.$$

The basic feature that connects between $\eta(\mathbf{u},\mathbf{v})$ and $\rho(\mathbf{u},\mathbf{v})$ is the [illegible]

$$\|A - \eta(\mathbf{u},\mathbf{v})\mathbf{u}\mathbf{v}^T\|_F^2 = \|A\|_F^2 - \Big(\rho(\mathbf{u},\mathbf{v})\Big)^2. \tag{5.10}$$

Since scaling of $\mathbf{u}$ and $\mathbf{v}$ does not affect this equality, there is no loss of generality in assuming that $\|\mathbf{u}\|_2 = 1$ and $\|\mathbf{v}\|_2 = 1$. In this case

$$\begin{aligned}\eta = \rho &= \mathbf{u}^T A\mathbf{v},\\ \|\mathbf{u}\mathbf{v}^T\|_F &= 1,\end{aligned}$$

and

$$\begin{aligned}\|A - \eta\mathbf{u}\mathbf{v}^T\|_F^2 &= \|A - \rho\mathbf{u}\mathbf{v}^T\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - \rho u_i v_j)^2\\ &= \sum_{i=1}^{m}\sum_{j=1}^{n}\Big[a_{ij}^2 - 2\rho a_{ij}u_i v_j + \rho^2(u_i v_j)^2\Big]\\ &= \|A\|_F^2 - 2\rho\mathbf{u}^T A\mathbf{v} + \rho^2\|\mathbf{u}\mathbf{v}^T\|_F^2\\ &= \|A\|_F^2 - 2\rho^2 + \rho^2 = \|A\|_F^2 - \rho^2,\end{aligned}$$

which proves (5.10). The meaning of the rectangular quotient equality is that solving the minimum norm problem (1.3) is equivalent to solving the problem

$$\text{maximize} \quad \rho(\mathbf{u},\mathbf{v}) = \mathbf{u}^T A\mathbf{v}/(\|\mathbf{u}\|_2\,\|\mathbf{v}\|_2). \tag{5.11}$$

An equivalent way to write the last problem is

$$\begin{aligned} &\text{maximize} \quad \sigma(\mathbf{u},\mathbf{v}) = \mathbf{u}^T A\mathbf{v} \\ &\text{subject to} \quad \|\mathbf{u}\|_2 = 1 \quad \text{and} \quad \|\mathbf{v}\|_2 = 1. \end{aligned} \tag{5.12}$$

By using the SVD of A (see (1.5)–(1.6)) the unit vectors in problem (5.12) can be expressed as

$$\mathbf{u} = \tilde{U}\tilde{\mathbf{u}}, \quad \|\tilde{\mathbf{u}}\|_2 \le 1, \quad \mathbf{v} = V\tilde{\mathbf{v}}, \quad \|\tilde{\mathbf{v}}\|_2 = 1, \tag{5.13}$$

where $\tilde{\mathbf{u}} = (\tilde{u}_1,\ldots,\tilde{u}_n)^T$, $\tilde{\mathbf{v}} = (\tilde{v}_1,\ldots,\tilde{v}_n)^T$, and $\tilde{U}$ is composed of the first n columns of U. This way, (5.12) is reduced to the problem

$$\begin{aligned} &\text{maximize} \quad \tilde{\sigma}(\tilde{\mathbf{u}},\tilde{\mathbf{v}}) = \sum_{i=1}^{n} \sigma_i \tilde{u}_i \tilde{v}_i \\ &\text{subject to} \quad \sum_{i=1}^{n} \tilde{u}_i^2 \le 1 \quad \text{and} \quad \sum_{i=1}^{n} \tilde{v}_i^2 = 1, \end{aligned} \tag{5.14}$$

whose solution is obtained at the points $\tilde{\mathbf{u}} = \tilde{\mathbf{v}} = \pm\mathbf{e}_1$. This shows that a pair of unit vectors, say, $\mathbf{u}^*$ and $\mathbf{v}^*$, solves (5.12) if and only if $\mathbf{u}^*$ and $\mathbf{v}^*$ constitute a dominant singular pair, and the corresponding optimal value is

$$\sigma(\mathbf{u}^*,\mathbf{v}^*) = \rho(\mathbf{u}^*,\mathbf{v}^*) = \sigma_1.$$

6. Retrieval of singular vectors. The question discussed in this section is how to retrieve a left singular vector from a right one, and vice versa. The aim of this note is to point out that the ultimate retrieval rule is both a minimization process and an orthogonalization process. This observation motivates the schemes proposed in the next sections. Further benefits of the retrieval rules are the error bounds derived at the end of this section.

Assume for a moment that $\mathbf{v}$ is a right singular vector of A that satisfies $A\mathbf{v} \neq \mathbf{0}$ and $\|\mathbf{v}\|_2 = 1$. Then the corresponding left singular vector $\mathbf{u}$ satisfies $A\mathbf{v} = \sigma\mathbf{u}$ and $\|\mathbf{u}\|_2 = 1$, where $\sigma > 0$ denotes the corresponding singular value of A. That is,

$$\mathbf{u} = A\mathbf{v}/\|A\mathbf{v}\|_2 \tag{6.1}$$

and

$$\sigma = \|A\mathbf{v}\|_2 = \mathbf{u}^T A\mathbf{v}. \tag{6.2}$$

In the general case $\mathbf{v}$ is just a given estimate for a right singular vector of A that satisfies $A\mathbf{v} \neq \mathbf{0}$. Nevertheless, as we are about to show, the above formulas remain useful for estimating the corresponding left singular vector and the corresponding singular value.

For this purpose we consider the problem

$$\text{minimize}\, F_\ell(\mathbf{u}) = \|A - \mathbf{u}\mathbf{v}^T\|_F^2, \tag{6.3}$$

where $\mathbf{u} = (u_1, \ldots, u_m)^T \in \mathbb{R}^m$ denotes the vector of unknowns. This problem is essentially a linear least squares problem whose solution is obtained by considering one unknown (one row) at a time. Let $\hat{\mathbf{u}} = (\hat{u}_1, \ldots, \hat{u}_m)^T \in \mathbb{R}^m$ denote the unique solution of this problem. Then, clearly,

$$\hat{u}_i = \mathbf{r}_i^T \mathbf{v} / \mathbf{v}^T \mathbf{v} \quad \text{for} \quad i = 1, \ldots, m, \tag{6.4}$$

where $\mathbf{r}_i^T$ denote the ith row of A. In other words,

$$\hat{\mathbf{u}} = A\mathbf{v} / \mathbf{v}^T \mathbf{v}. \tag{6.5}$$

Let $\hat{\mathbf{r}}_i^T$ denote the ith row of the matrix $\hat{A} = A - \hat{\mathbf{u}}\mathbf{v}^T$. Then $\hat{\mathbf{r}}_i^T = \mathbf{r}_i^T - \hat{u}_i \mathbf{v}^T$ is obtained by orthogonalizing $\mathbf{r}_i$ against $\mathbf{v}$. Consequently $\hat{\mathbf{r}}_i^T \mathbf{v} = 0$ and $\hat{A}\mathbf{v} = \mathbf{0}$. Of course, if $\mathbf{v}$ and $\hat{\mathbf{u}}$ are normalized to be unit vectors, then (6.5) coincides with (6.1).

A similar approach enables us to derive a right singular vector from a left one. Assume first that $\mathbf{u}$ is a left singular vector of A that satisfies $A^T\mathbf{u} \neq \mathbf{0}$ and $\|\mathbf{u}\|_2 = 1$. Then the corresponding right singular vector $\mathbf{v}$ and the corresponding singular value σ must satisfy

$$\mathbf{v} = A^T \mathbf{u} / \|A^T \mathbf{u}\|_2 \tag{6.6}$$

and

$$\sigma = \|A^T \mathbf{u}\|_2 = \mathbf{u}^T A \mathbf{v}. \tag{6.7}$$

In the general case, when $\mathbf{u}$ is just an estimate for a left singular vector, the last equalities provide estimates of the corresponding right singular vector $\mathbf{v}$ and the corresponding singular value. To see this point we consider the minimum norm problem

$$\text{minimize} \quad F_r(\mathbf{v}) = \|A - \mathbf{u}\mathbf{v}^T\|_F^2, \tag{6.8}$$

where here $\mathbf{v} = (v_1, \ldots, v_n)^T \in \mathbb{R}^n$ denotes the vector of unknowns. In this case the solution vector

$$\hat{\mathbf{v}} = A^T \mathbf{u} / \mathbf{u}^T \mathbf{u} \tag{6.9}$$

is obtained by orthogonalizing the columns of A against $\mathbf{u}$.

The retrieval formulas enable us to derive simple error bounds that resemble (4.7). Let $\hat{\mathbf{v}} \in \mathbb{R}^n$ be a given unit vector that satisfies $A\hat{\mathbf{v}} \neq \mathbf{0}$, and let

$$\hat{\mathbf{u}} = A\hat{\mathbf{v}} / \|A\hat{\mathbf{v}}\|_2$$

and

$$\hat{\sigma} = \|A\hat{\mathbf{v}}\|_2 = \hat{\mathbf{u}}^T A \hat{\mathbf{v}}$$

provide the corresponding estimates of a left singular vector and a singular value, respectively. Below we derive a bound on the distance between $\hat{\sigma}$ and a singular value of A. Recall that the squares of the singular values of A are the eigenvalues of $A^T A$. Note also that $\hat{\sigma}^2 = \hat{\mathbf{v}}^T A^T A \hat{\mathbf{v}}$ is the symmetric Rayleigh quotient of $A^T A$ related to $\hat{\mathbf{v}}$. Thus by using (4.7) with $S = A^T A$ we conclude the existence of a singular value of A, $\sigma \geq 0$, such that

$$|\sigma^2 - \hat{\sigma}^2| \leq \|A^T A \hat{\mathbf{v}} - \hat{\sigma}^2 \mathbf{v}\|_2.$$

The last inequality can be rewritten as

$$|\sigma - \hat{\sigma}| \cdot |\sigma + \hat{\sigma}| \leq \hat{\sigma}\|A^T\hat{\mathbf{u}} - \hat{\sigma}\hat{\mathbf{v}}\|_2,$$

which leads to

$$|\sigma - \hat{\sigma}| \leq |\hat{\sigma}/(\sigma + \hat{\sigma})| \cdot \|A^T\hat{\mathbf{u}} - \hat{\sigma}\hat{\mathbf{v}}\|_2 \leq \|A^T\hat{\mathbf{u}} - \hat{\sigma}\hat{\mathbf{v}}\|_2.$$

That is,

$$|\sigma - \hat{\sigma}| \leq \|A^T\hat{\mathbf{u}} - \hat{\sigma}\hat{\mathbf{v}}\|_2. \tag{6.10}$$

The treatment of a left singular vector is done in a similar way. Let $\tilde{\mathbf{u}} \in \mathbb{R}^m$ be a given unit vector that satisfies $A^T\tilde{\mathbf{u}} \neq \mathbf{0}$, and let

$$\tilde{\mathbf{v}} = A^T\tilde{\mathbf{u}}/\|A^T\tilde{\mathbf{u}}\|_2$$

and

$$\tilde{\sigma} = \|A^T\tilde{\mathbf{u}}\|_2 = \tilde{\mathbf{u}}^T A\tilde{\mathbf{v}}$$

denote the corresponding estimates of a right singular vector and a singular value, respectively. Then similar arguments prove the existence of a singular value of A, $\sigma \geq 0$, such that

$$|\sigma - \tilde{\sigma}| \leq \|A\tilde{\mathbf{v}} - \tilde{\sigma}\tilde{\mathbf{u}}\|_2. \tag{6.11}$$

7. Rectangular iterations. In this section we present a simple iterative algorithm for solving the minimum norm problem

$$\text{minimize } F(\mathbf{u}, \mathbf{v}) = \|A - \mathbf{u}\mathbf{v}^T\|_F^2. \tag{7.1}$$

The kth iteration, $k = 1, 2, 3, \ldots$, starts with $\mathbf{u}_{k-1}$ and $\mathbf{v}_{k-1}$ and ends with $\mathbf{u}_k$ and $\mathbf{v}_k$. Given $\mathbf{v}_{k-1}$, the vector $\mathbf{u}_k$ is obtained by solving the problem

$$\text{minimize } \varphi(\mathbf{u}) = \|A - \mathbf{u}\mathbf{v}_{k-1}^T\|_F^2.$$

That is,

$$\mathbf{u}_k = A\mathbf{v}_{k-1}/\mathbf{v}_{k-1}^T\mathbf{v}_{k-1}. \tag{7.2}$$

Then $\mathbf{v}_k$ is obtained by solving the problem

$$\text{minimize } \Psi(\mathbf{v}) = \|A - \mathbf{u}_k\mathbf{v}^T\|_F^2,$$

which gives

$$\mathbf{v}_k = A^T\mathbf{u}_k/\mathbf{u}_k^T\mathbf{u}_k. \tag{7.3}$$

Observe that the iteration (7.2)–(7.3) is essentially the retrieval rules (6.5) and (6.9). Note also that minimizing $F(\mathbf{u}, \mathbf{v})$ by changing one variable at a time (in the order $u_1, \ldots, u_m, v_1, \ldots, v_n$) results in the same basic iteration. Furthermore, let

$$\hat{\mathbf{v}}_k = \mathbf{v}_k/\|\mathbf{v}_k\|_2, \quad k = 0, 1, 2, \ldots,$$

denote the corresponding sequence of unit vectors. Let the sequence $\tilde{\mathbf{v}}_k, k = 0, 1, 2, \ldots,$ be generated by applying the power method to A^TA, with $\tilde{\mathbf{v}}_0 = \hat{\mathbf{v}}_0$ as the starting point. Then, in exact arithmetic,

$$\tilde{\mathbf{v}}_k = \hat{\mathbf{v}}_k \quad \text{for} \quad k = 1, 2, \ldots .$$

Similarly, the sequence

$$\hat{\mathbf{u}}_k = \mathbf{u}_k / \|\mathbf{u}_k\|_2, \quad k = 1, 2, \ldots,$$

is obtained by applying the power method to AA^T, with $\hat{\mathbf{u}}_1$ as the starting point.

Replacing the order of $\mathbf{u}$ and $\mathbf{v}$ results in the iteration

$$\mathbf{v}_k = A^T\mathbf{u}_{k-1} / \mathbf{u}_{k-1}^T\mathbf{u}_{k-1} \tag{7.4}$$

and

$$\mathbf{u}_k = A\mathbf{v}_k / \mathbf{v}_k^T\mathbf{v}_k, \tag{7.5}$$

which satisfies similar relations with the power method applied to AA^T. In order to distinguish between the two schemes we introduce the following terminology. The scheme (7.2)–(7.3) is called the "[illegible]," while (7.4)–(7.5) is called the "[illegible]." The reason for these names and the difference between the two schemes are explained in the next section. Both iterations satisfy the equality

$$(\mathbf{u}_k^T\mathbf{u}_k)(\mathbf{v}_k^T\mathbf{v}_k) = \mathbf{u}_k^TA\mathbf{v}_k, \tag{7.6}$$

so the corresponding rectangular quotients satisfy

$$\eta_k \equiv \eta(\mathbf{u}_k, \mathbf{v}_k) = 1 \tag{7.7}$$

and

$$\rho_k \equiv \rho(\mathbf{u}_k, \mathbf{v}_k) = \|\mathbf{u}_k\|_2 \, \|\mathbf{v}_k\|_2. \tag{7.8}$$

Combining these relations with the rectangular quotient equality shows that

$$\|A - \mathbf{u}_k\mathbf{v}_k^T\|_F^2 = \|A\|_F^2 - (\mathbf{u}_k^T\mathbf{u}_k)(\mathbf{v}_k^T\mathbf{v}_k). \tag{7.9}$$

Therefore, since the sequence $\{\|A - \mathbf{u}_k\mathbf{v}_k^T\|_F^2\}$ decreases monotonously, the sequences $\{(\mathbf{u}_k^T\mathbf{u}_k)(\mathbf{v}_k^T\mathbf{v}_k)\}$ and $\{\rho_k\}$ increase monotonously. Moreover, assume for simplicity that $\sigma_1 > \sigma_2$, and let $\mathbf{u}^*$ and $\mathbf{v}^*$ denote the corresponding left and right dominant singular vectors, where $\|\mathbf{u}^*\|_2 = 1$ and $\|\mathbf{v}^*\|_2 = 1$. Then the close links with the power method imply that

$$\lim_{k\to\infty} \hat{\mathbf{u}}_k = \mathbf{u}^* \tag{7.10}$$

and

$$\lim_{k\to\infty} \hat{\mathbf{v}}_k = \mathbf{v}^* \tag{7.11}$$

and that these sequences converge at a linear rate, proportional to σ_2^2/σ_1^2. A further use of (7.6) shows that in left iterations

$$(\mathbf{u}_k^T\mathbf{u}_k)(\mathbf{v}_k^T\mathbf{v}_k) = \hat{\mathbf{u}}_k^TAA^T\hat{\mathbf{u}}_k,$$

while in right iterations

$$(\mathbf{u}_k^T\mathbf{u}_k)(\mathbf{v}_k^T\mathbf{v}_k) = \hat{\mathbf{v}}_k^TA^TA\hat{\mathbf{v}}_k.$$

Thus in both cases

$$\lim_{k\to\infty} \rho_k^2 = \lim_{k\to\infty} (\mathbf{u}_k^T \mathbf{u}_k)(\mathbf{v}_k^T \mathbf{v}_k) = \sigma_1^2$$

and

$$\lim_{k\to\infty} \rho_k = \sigma_1, \tag{7.12}$$

where the sequence $\{\rho_k\}$ converges at a linear rate, proportional to σ_2^2/σ_1^2.

Recall, however, that the initial rate of convergence is expected to be faster than the asymptotic rate: If the starting point $\mathbf{u}_0$ (or $\mathbf{v}_0$) is not nearly perpendicular to $\mathbf{u}^*$ (or $\mathbf{v}^*$), then a few rectangular iterations are likely to provide a fair estimate of a dominant singular triplet. If no preliminary information is available, then taking $\mathbf{u}_0$ (or $\mathbf{v}_0$) to be the column (or row) of A that has the largest Euclidean norm is a reasonable choice (see the next section). Note also that the error bounds (6.10) and (6.11) provide useful stopping conditions.

8. Orthogonalization via deflation. In this section we present a new orthogonalization method that is called orthogonalization via deflation. By starting from $A_1 = A$ the new method generates a sequence of matrices $A_1, A_2, A_3, \ldots$, where $A_{\ell+1}$ is obtained from A_ℓ by the rule

$$A_{\ell+1} = A_\ell - \tilde{\sigma}_\ell \tilde{\mathbf{u}}_\ell \tilde{\mathbf{v}}_\ell^T, \quad \ell = 1, 2, \ldots, \tag{8.1}$$

where $\tilde{\mathbf{u}}_\ell \in \mathbb{R}^m$ and $\tilde{\mathbf{v}}_\ell \in \mathbb{R}^n$ are unit vectors and $\tilde{\sigma}_\ell > 0$ is a positive number that estimates the corresponding singular value. Below, we describe three basic versions of the method, which differ in the definition of the rank-one matrix $\tilde{\sigma}_\ell \tilde{\mathbf{u}}_\ell \tilde{\mathbf{v}}_\ell^T$.

. Here $\tilde{\mathbf{u}}_\ell$ is an arbitrary unit vector from Range(A_ℓ),

$$\tilde{\mathbf{v}}_\ell = A_\ell^T \tilde{\mathbf{u}}_\ell / \|A_\ell^T \tilde{\mathbf{u}}_\ell\|_2, \quad \text{and} \quad \tilde{\sigma}_\ell = (\tilde{\mathbf{u}}_\ell)^T A_\ell \tilde{\mathbf{v}}_\ell. \tag{8.2}$$

. Here $\tilde{\mathbf{v}}_\ell$ is an arbitrary unit vector from Range(A_ℓ^T),

$$\tilde{\mathbf{u}}_\ell = A_\ell \tilde{\mathbf{v}}_\ell / \|A_\ell \tilde{\mathbf{v}}_\ell\|_2, \quad \text{and} \quad \tilde{\sigma}_\ell = (\tilde{\mathbf{u}}_\ell)^T A_\ell \tilde{\mathbf{v}}_\ell. \tag{8.3}$$

. Let $\hat{\mathbf{u}}_\ell$ and $\hat{\mathbf{v}}_\ell$ be an arbitrary pair of unit vectors that satisfy

$$\hat{\mathbf{u}}_\ell \in \text{Range}(A_\ell), \quad \hat{\mathbf{v}}_\ell \in \text{Range}(A_\ell^T), \qquad \text{and} \quad \hat{\mathbf{u}}_\ell^T A_\ell \hat{\mathbf{v}}_\ell > 0. \tag{8.4}$$

Then here

$$A_{\ell+1} = A_\ell - (A_\ell \hat{\mathbf{v}}_\ell)(A_\ell^T \hat{\mathbf{u}}_\ell)^T / (\hat{\mathbf{u}}_\ell^T A_\ell \hat{\mathbf{v}}_\ell). \tag{8.5}$$

That is, here (8.1) is carried out with

$$\tilde{\mathbf{u}}_\ell = A_\ell \hat{\mathbf{v}}_\ell / \|A_\ell \hat{\mathbf{v}}_\ell\|_2,$$

$$\tilde{\mathbf{v}}_\ell = A_\ell^T \hat{\mathbf{u}}_\ell / \|A_\ell^T \hat{\mathbf{u}}_\ell\|_2,$$

and

$$\tilde{\sigma}_\ell = (\|A_\ell \hat{\mathbf{v}}_\ell\|_2 \|A_\ell^T \hat{\mathbf{u}}_\ell\|_2) / (\hat{\mathbf{u}}_\ell^T A_\ell \hat{\mathbf{v}}_\ell).$$

To analyze the proposed methods we introduce the following notations. Let $\tilde{U}_\ell$ denote the $m \times \ell$ matrix whose columns are $\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_\ell$. Let $\tilde{V}_\ell$ denote the $n \times \ell$ matrix whose columns are $\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_\ell$. Let $\tilde{D}_\ell$ denote a diagonal $\ell \times \ell$ matrix whose diagonal entries are $\tilde{\sigma}_1, \ldots, \tilde{\sigma}_\ell$. That is,

$$\tilde{U}_\ell = [\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_\ell], \quad \tilde{V}_\ell = [\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_\ell], \quad \text{and} \quad \tilde{D}_\ell = \operatorname{diag}\{\tilde{\sigma}_1, \ldots, \tilde{\sigma}_\ell\}. \tag{8.6}$$

Then, at the end of the ℓth deflation stage,

$$A_{\ell+1} = A - \sum_{j=1}^{\ell} \tilde{\sigma}_j \tilde{\mathbf{u}}_j (\tilde{\mathbf{v}}_j)^T = A - \tilde{U}_\ell \tilde{D}_\ell \tilde{V}_\ell^T = A - \tilde{B}_\ell, \tag{8.7}$$

where

$$\tilde{B}_\ell = \sum_{j=1}^{\ell} \tilde{\sigma}_j \tilde{\mathbf{u}}_j (\tilde{\mathbf{v}}_j)^T = \tilde{U}_\ell \tilde{D}_\ell \tilde{V}_\ell^T \tag{8.8}$$

may serve as a low-rank approximation of A. The differences between the three schemes are explained below. Yet, as we shall see, the three schemes share the "exactness property"

$$A = \tilde{U}_r \tilde{D}_r \tilde{V}_r^T, \tag{8.9}$$

where $r = \operatorname{rank}(A)$. The titles of the theorems refer to the rule which defines the triplet $\{\tilde{\sigma}_\ell, \tilde{\mathbf{u}}_\ell, \tilde{\mathbf{v}}_\ell\}$.

THEOREM 2 (left-side orthogonalization). [illegible] $\ell = 1, \ldots, r$

$$\operatorname{Range}(A_{\ell+1}) \subseteq \operatorname{Range}(A_\ell), \tag{8.10}$$

$$\operatorname{Range}(\tilde{U}_\ell) \subseteq \operatorname{Range}(A), \tag{8.11}$$

$$\tilde{U}_\ell^T A_{\ell+1} = 0, \tag{8.12}$$

[illegible]

$$\tilde{U}_\ell^T \tilde{U}_\ell = I. \tag{8.13}$$

[illegible] $\ell = r$ [illegible] $\tilde{U}_r$ [illegible] Range(A),

$$A_{r+1} = 0, \tag{8.14}$$

[illegible]

$$A = \tilde{U}_r \tilde{D}_r \tilde{V}_r^T. \tag{8.15}$$

[illegible]. The proof is by induction on ℓ. For $\ell = 1$ the relations (8.10)–(8.13) are direct consequences of the definition of $\{\tilde{\sigma}_1, \tilde{\mathbf{u}}_1, \tilde{\mathbf{v}}_1\}$. Assume now that (8.10)–(8.13) hold for $\ell - 1$. That is,

$$\operatorname{Range}(A_\ell) \subseteq \operatorname{Range}(A),$$
$$\operatorname{Range}(\tilde{U}_{\ell-1}) \subseteq \operatorname{Range}(A),$$
$$\tilde{U}_{\ell-1}^T A_\ell = 0,$$

and

$$U_{\ell-1}^T \tilde{U}_{\ell-1} = I.$$

On the other hand, the definition of the triplet $\{\tilde{\sigma}_\ell, \tilde{\mathbf{u}}_\ell, \tilde{\mathbf{v}}_\ell\}$ implies the relations

$$\tilde{\mathbf{u}}_\ell \in \mathrm{Range}(A_\ell) \subseteq \mathrm{Range}(A), \quad \tilde{\mathbf{u}}_\ell^T \tilde{\mathbb{U}}_{\ell-1} = \mathbf{0}, \quad \text{and} \quad \tilde{\mathbf{u}}_\ell^T A_{\ell+1} = \mathbf{0}.$$

So combining these relations with the induction assumptions yields (8.10)–(8.13). □

Recall that Gram–Schmidt orthogonalization has three basic versions: classical Gram–Schmidt, a column-oriented version of modified Gram–Schmidt (MGS), and a row-oriented version of MGS. The last version is the only one that is able to incorporate column pivoting. Yet, in exact arithmetic and without column pivoting, all of these versions produce the same QR factorization of A, which is closely related to Householder QR factorization of the $(m+n)\times n$ matrix $\binom{0}{A}$. For a detailed discussion of these methods see [3], [6], [7], [14], [20], [28].

COROLLARY 3 (relation to Gram–Schmidt orthogonalization). [illegible] $\tilde{\mathbf{u}}_\ell$ [illegible] A_ℓ [illegible] $\mathbf{c}_j$, $j = 1, \ldots, n$, [illegible] j [illegible] A_ℓ [illegible] j^* [illegible]

$$\|\mathbf{c}_{j^*}\|_2 = \max\{\|\mathbf{c}_1\|_2, \ldots, \|\mathbf{c}_n\|_2\}.$$

[illegible]

$$\tilde{\mathbf{u}}_\ell = \mathbf{c}_{j^*} / \|\mathbf{c}_{j^*}\|_2. \tag{8.16}$$

[illegible] A_ℓ [illegible] $\mathbf{c}_\ell = \mathbf{c}_{j^*}$ [illegible] (8.1)–(8.2) [illegible] A [illegible]. By using induction on ℓ it is easy to verify that the first ℓ columns of $A_{\ell+1}$ are null vectors, while the other columns of $A_{\ell+1}$ are obtained by orthogonalizing the columns of A_ℓ against the ℓth column of A_ℓ. Consequently the first $\ell-1$ entries of $\tilde{\mathbf{v}}_\ell$ are zeros, while the other entries are defined by the corresponding orthogonalization factors. □

THEOREM 4 (right-side orthogonalization). [illegible] $\ell = 1, \ldots, r$

$$\mathrm{Range}(A_{\ell+1}^T) \subseteq \mathrm{Range}(A_\ell^T), \tag{8.17}$$

$$\mathrm{Range}(\tilde{V}_\ell) \subseteq \mathrm{Range}(A^T), \tag{8.18}$$

$$A_{\ell+1}\tilde{V}_\ell = 0, \tag{8.19}$$

[illegible]

$$\tilde{V}_\ell^T \tilde{V}_\ell = I. \tag{8.20}$$

[illegible] $\ell = r$ [illegible] $\tilde{V}_r$ [illegible] $\mathrm{Range}(A^T)$,

$$A_{r+1} = 0, \tag{8.21}$$

[illegible]

$$A = \tilde{U}_r \tilde{D}_r \tilde{V}_r^T. \tag{8.22}$$

[illegible]. The proof is achieved by using induction on ℓ, as in Theorem 2. □

COROLLARY 5 (relation to Gram–Schmidt orthogonalization). $\tilde{\mathbf{v}}_\ell$ A_ℓ A^T

THEOREM 6 (two-sides orthogonalization). $\hat{U}_\ell \in \mathbb{R}^{m\times\ell}$, $\hat{V}_\ell \in \mathbb{R}^{n\times\ell}$, $\hat{D}_\ell \in \mathbb{R}^{\ell\times\ell}$

$$\hat{U}_\ell = [\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_\ell], \quad \hat{V}_\ell = [\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_\ell], \quad \text{and} \quad \hat{D}_\ell = \operatorname{diag}\{\hat{\sigma}_1, \ldots, \hat{\sigma}_\ell\}, \tag{8.23}$$

$$\hat{\sigma}_j = \hat{\mathbf{u}}_j^T A \hat{\mathbf{v}}_j \quad \text{for} \quad j = 1, \ldots, r. \tag{8.24}$$

$\ell = 1, \ldots, r$

$$\operatorname{Range}(A_{\ell+1}) \subseteq \operatorname{Range}(A_\ell), \quad \operatorname{Range}(A_{\ell+1}^T) \subseteq \operatorname{Range}(A_\ell^T), \tag{8.25}$$

$$\operatorname{Range}(\hat{U}_\ell) \subseteq \operatorname{Range}(A), \quad \operatorname{Range}(\hat{V}_\ell) \subseteq \operatorname{Range}(A^T), \tag{8.26}$$

$$\hat{U}_\ell^T A_{\ell+1} = 0, \qquad A_{\ell+1}\hat{V}_\ell = 0, \tag{8.27}$$

$$\hat{U}_\ell^T \hat{U}_\ell = I, \quad \text{and} \quad \hat{V}_\ell^T \hat{V}_\ell = I. \tag{8.28}$$

$\ell = r$ $\hat{U}_r$ $\hat{V}_r$ $\operatorname{Range}(A)$ $\operatorname{Range}(A^T)$

$$A_{r+1} = 0 \tag{8.29}$$

$$A = \tilde{U}_r \tilde{D}_r \tilde{V}_r. \tag{8.30}$$

. The proof is by induction on ℓ. The key observation is that here the deflation step (8.5) implies that

$$A_{\ell+1}^T \hat{\mathbf{u}}_\ell = \mathbf{0} \quad \text{and} \quad A_{\ell+1}\hat{\mathbf{v}}_\ell = \mathbf{0}. \tag{8.31}$$

From this point, continue as in the proof of Theorem 2. ∎

The quality of the resulting factorizations, as substitutes for the SVD, depends on the quality of the deflating triplets $\{\tilde{\sigma}_\ell, \tilde{\mathbf{u}}_\ell, \tilde{\mathbf{v}}_\ell\}$, as substitutes for a dominant triplet of A_ℓ. As we have seen, a small number of rectangular iterations is likely to provide a fair estimate of a dominant triplet of A_ℓ. Furthermore, the relation to pivoted MGS suggests a natural starting point for the rectangular iterations: Take a column (or row) of A_ℓ that has the largest Euclidean norm. Note also that the error bounds (6.10) and (6.11) provide useful stopping conditions for the rectangular iterations.

The two-sides orthogonalization process (8.4)–(8.5) constructs two low-rank approximations of A:

$$\tilde{B}_\ell = \tilde{U}_\ell \tilde{D}_\ell \tilde{V}_\ell^T = \sum_{j=1}^{\ell} \tilde{\sigma}_j \tilde{\mathbf{u}}_j \tilde{\mathbf{v}}_j^T \tag{8.32}$$

and

$$\hat{B}_\ell = \hat{U}_\ell \hat{D}_\ell \hat{V}_\ell^T = \sum_{j=1}^{\ell} \hat{\sigma}_j \hat{\mathbf{u}}_j \hat{\mathbf{v}}_j^T. \tag{8.33}$$

The second decomposition has the advantage that both $\hat{U}_\ell$ and $\hat{V}_\ell$ have orthonormal columns, which makes it a valuable substitute for the SVD's low-rank approximation. The definition (8.24) of the diagonal matrix $\hat{D}_\ell$ is motivated by the observation that the diagonal entries of this matrix solve the minimum norm problem

$$\text{minimize} \quad F(\sigma_1, \ldots, \sigma_\ell) = \left\| A - \sum_{j=1}^{\ell} \sigma_j \hat{\mathbf{u}}_j \hat{\mathbf{v}}_j^T \right\|_F^2 . \tag{8.34}$$

Of course, if one is interested only in (8.33), then there is no need to store the matrices $\tilde{U}_\ell$ and $\tilde{V}_\ell$.

As noted by one referee, the two-sides orthogonalization process (8.4)–(8.5) is a special case of a more general method, the Wedderburn–Egervary rank-reducing (WERR) process, e.g., [4], [12], [31]. By using our notations the ℓth step of a WERR process has the form

$$A_{\ell+1} = A_\ell - (A_\ell \hat{\mathbf{v}}_\ell)(A_\ell^T \hat{\mathbf{u}}_\ell)^T / (\hat{\mathbf{u}}_\ell^T A_\ell \hat{\mathbf{v}}_\ell), \quad \ell = 1, 2, \ldots, \tag{8.35}$$

where the vectors $\hat{\mathbf{u}}_\ell \in \mathbb{R}^m$ and $\hat{\mathbf{v}}_\ell \in \mathbb{R}^n$ need to satisfy

$$\hat{\mathbf{u}}_\ell^T A_\ell \hat{\mathbf{v}}_\ell \neq 0. \tag{8.36}$$

The name "rank-reducing process" comes from the observation that

$$\operatorname{rank}(A_{\ell+1}) = \operatorname{rank}(A_\ell) - 1, \tag{8.37}$$

which implies the "finite termination" property $A_{r+1} = 0$. It is illustrated in [4] that several important matrix factorizations belong to this class of methods. However, to the best of our knowledge, the two-sides orthogonalization process (8.4)–(8.5) is a new method, distinguished by the orthogonality of the matrices $\hat{U}_\ell$ and $\hat{V}_\ell$, which is gained by replacing (8.36) with (8.4). That is, orthogonality is gained by choosing the vectors $\hat{\mathbf{u}}_\ell$ and $\hat{\mathbf{v}}_\ell$ from $\operatorname{Range}(A_\ell)$ and $\operatorname{Range}(A_\ell^T)$, respectively. Another feature that characterizes our approach is the use of rectangular iterations to estimate a dominant pair of singular vectors and to use the computed vectors in (8.35). It is the incorporation of these two properties that makes the resulting decomposition a valuable substitute for the SVD. Similar remarks apply to left-side and right-side orthogonalizations, although the relation of these methods to the WERR process (8.35) is somewhat less obvious.

The treatment of rounding errors is done as in the symmetric case, by using singular values instead of eigenvalues. To simplify the coming discussion we concentrate on the left-side process (8.2), assuming that each triplet $\{\rho_\ell, \mathbf{u}_\ell, \mathbf{v}_\ell\}$ provides a fair estimate of a dominant triplet of A_ℓ. The last assumption means that the size of $\|A_\ell\|_F$ is about $\sigma_\ell + \cdots + \sigma_n$. On the other hand, in floating point arithmetic the computed matrix A_ℓ contains an error matrix E_ℓ whose Frobenius norm is about $\varepsilon(\sigma_1 + \cdots + \sigma_{\ell-1})$, where ε denotes the machine precision in our computations. That is,

$$\|E_{\ell+1}\|_F / \|A_{\ell+1}\|_F \approx \varepsilon(\sigma_1 + \cdots + \sigma_\ell) / (\sigma_{\ell+1} + \cdots + \sigma_n). \tag{8.38}$$

Thus, when A is an ill-conditioned matrix, small singular values are computed with a large relative error. Nevertheless, the overall perturbation of the singular values is small: Let $\tilde{\sigma}_1, \ldots, \tilde{\sigma}_n$ denote the singular values of A_ℓ, and let $\hat{\sigma}_1, \ldots, \hat{\sigma}_n$ denote those of $A_\ell - E_\ell$. Then from Weyl's theorem [28, p. 69] one obtains

$$|\hat{\sigma}_i - \tilde{\sigma}_i| \leq \|E_\ell\|_2 \leq \|E_\ell\|_F \approx \varepsilon(\sigma_1 + \cdots + \sigma_{\ell-1}), \quad i = 1, \ldots, n. \tag{8.39}$$

A further difficulty that arises as the ratio (8.38) grows is the gradual loss of orthogonality. Yet, as in the symmetric case, this difficulty is easily resolved by applying reorthogonalization. For example, in the ℓth step of a left-orthogonalization process, $\mathbf{u}_\ell$ is orthogonalized against $\mathbf{u}_1, \ldots, \mathbf{u}_{\ell-1}$. Similarly, in the ℓth step of a right-orthogonalization process, $\mathbf{v}_\ell$ is orthogonalized against $\mathbf{v}_1, \ldots, \mathbf{v}_{\ell-1}$, and so forth.

9. Missing data estimation. In this section we discuss the case when some entries of the data matrix A are unknown. In such a situation it is often desired to construct a low-rank estimate of A in spite of the missing data, e.g., [1], [2], [13], [15], [30]. The minimum norm approach is easily adapted to handle this difficulty. Consider, for example, the rectangular minimum norm problem (7.1) when some entries of A are missing. In this case the problem to be solved is redefined as

$$\text{minimize} \quad F(\mathbf{u}, \mathbf{v}) = \sum_{(i,j)\in\mathbb{K}} (a_{ij} - u_i v_j)^2, \tag{9.1}$$

where

$$\mathbb{K} = \{(i,j) \,|\, a_{ij} \text{ is known}\}.$$

That is, the sum in (9.1) is restricted to known entries of A. Let the set

$$\mathbb{C}_j = \{i \,|\, a_{ij} \text{ is known}\}$$

contain the row indices of known entries in the jth column of A, $j = 1, \ldots, n$. Then a second way to write (9.1) is

$$\text{minimize} \quad F(\mathbf{u}, \mathbf{v}) = \sum_{j=1}^{n} \sum_{i\in\mathbb{C}_j} (a_{ij} - u_i v_j)^2. \tag{9.2}$$

Similarly, let

$$\mathbb{R}_i = \{j \,|\, a_{ij} \text{ is known}\}$$

contain the column indices of known entries in the ith row of A, $i = 1, \ldots, m$. Then a third way to write (9.1) is

$$\text{minimize} \quad F(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{m} \sum_{j\in\mathbb{R}_j} (a_{ij} - u_i v_j)^2. \tag{9.3}$$

The adaptation of the "left iteration" (7.2)–(7.3) to solve (9.1) is done in the following way. As before, the kth iteration, $k = 1, 2, 3, \ldots$, starts with $\mathbf{u}_{k-1}$ and $\mathbf{v}_{k-1}$ and ends with $\mathbf{u}_k$ and $\mathbf{v}_k$. Given $\mathbf{v}_{k-1} = (\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_n)^T$, the vector $\mathbf{u}_k = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_m)^T$ is obtained by solving the problem

$$\text{minimize} \quad \varphi(\mathbf{u}) = \sum_{i=1}^{m} \sum_{j\in\mathbb{R}_i} (a_{ij} - u_i \tilde{v}_j)^2.$$

That is,

$$\hat{u}_i = \left(\sum_{j\in\mathbb{R}_i} a_{ij}\tilde{v}_j \right) \Big/ \left(\sum_{j\in\mathbb{R}_i} \tilde{v}_j^2 \right) \quad \text{for} \quad i = 1, \ldots, m. \tag{9.4}$$

Then $\mathbf{v}_k = (\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_n)^T$ is obtained by solving the problem

$$\text{minimize} \quad \Psi(\mathbf{v}) = \sum_{j=1}^{n} \sum_{i \in \mathbb{C}_j} (a_{ij} - \hat{u}_i v_j)^2.$$

That is,

$$\hat{v}_j = \left(\sum_{i \in \mathbb{C}_j} a_{ij} \hat{u}_i \right) \Big/ \left(\sum_{i \in \mathbb{C}_j} \hat{u}_i^2 \right) \quad \text{for} \quad j = 1, \ldots, n. \tag{9.5}$$

The "right iteration" (7.4)–(7.5) is adapted in a similar way.

The ability to solve (9.1) enables us to apply the deflation by subtraction process when some entries of A are unknown. The ℓth stage of this process, $\ell = 1, 2, \ldots$, needs only minor changes. Let $a_{ij}^{(\ell)}$ denote the (i, j) component of A_ℓ. Then the problem

$$\text{minimize} \quad F_\ell(\mathbf{u}, \mathbf{v}) = \sum_{(i,j) \in \mathbb{K}} (a_{ij}^{(\ell)} - u_i v_j)^2 \tag{9.6}$$

is solved by the modified iterations described above. Let $\mathbf{u}^*$ and $\mathbf{v}^*$ denote the computed solution of (9.6). Then $A_{\ell+1}$ is obtained from A_ℓ by the rule

$$A_{\ell+1} = A_\ell - \mathbf{u}^*(\mathbf{v}^*)^T = A_\ell - \rho_\ell^* \mathbf{u}_\ell^* (\mathbf{v}_\ell^*)^T, \tag{9.7}$$

where

$$\mathbf{u}_\ell^* = \mathbf{u}^* / \|\mathbf{u}^*\|_2, \quad \mathbf{v}_\ell^* = \mathbf{v}^* / \|\mathbf{v}^*\|_2, \quad \text{and} \quad \rho_\ell^* = \|\mathbf{u}^*\|_2 \, \|\mathbf{v}^*\|_2.$$

Note that the entries of $A_{\ell+1}$ which correspond to unknown entries of A are still considered as unknown. Note also that the sequence

$$\xi_\ell = \sum_{(i,j) \in \mathbb{K}} (a_{ij}^{(\ell)})^2, \qquad \ell = 1, 2, \ldots, \tag{9.8}$$

decreases monotonously. Thus, once we reach an iteration index ℓ^* for which ξ_{ℓ^*} is considerably smaller than ξ_1, the sum

$$B_{\ell^*} = \sum_{\ell=1}^{\ell^*} \rho_\ell^* \mathbf{u}_\ell^* (\mathbf{v}_\ell^*)^T \tag{9.9}$$

may serve as a low-rank approximation of A.

10. Concluding remarks. The power method, the deflation by subtraction process, and the Rayleigh quotient are among the early tools for solving symmetric eigenvalue problems. It is interesting, however, to ask whether these methods and concepts can be extended for calculating the SVD of rectangular matrices. The minimum norm approach that we introduce gives a definite answer to this question. At the same time it reveals new features of these tools and paves the way for modified schemes.

The new deflation by subtraction process for symmetric matrices has clear advantages over the classical method of Hotelling [16], [17], namely, the "finite termination" property, the "exactness" property, the "orthogonality" property, and the fact that

these features hold regardless the quality of the computed eigenpairs. One aim of the paper is to point out that a small number of power iterations per eigenpair are sufficient to provide a meaningful characterization of the eigensystem.

The symmetric quotient equality (4.8) is a new useful relation that directly connects the minimum norm problem (1.1) and the Rayleigh quotient, showing that minimizing (1.1) is equivalent to maximizing the Rayleigh quotient. An extended version of this equality sheds new light on Key Fan's maximum principle; see [9]. The ability to find a dominant eigenpair by solving (1.1) opens the door for effective minimization techniques. The simple relaxation method proposed in [8] illustrates this point.

When moving to rectangular matrices, singular values take the role of eigenvalues, while the rectangular Rayleigh quotient (5.9) replaces the Rayleigh quotient (4.4). The minimum norm approach enables us to see the similarity between the two cases. The rectangular quotient equality (5.10) directly connects the minimum norm problem (1.3) and the rectangular Rayleigh quotient (5.9), showing that minimizing (1.3) is equivalent to maximizing the rectangular Rayleigh quotient. In [9] we establish an extended version of this equality, which adds new insight into the Eckart–Young theorem.

At this point it is instructive to see the difference between the rectangular Rayleigh quotient (5.9) and the generalized Rayleigh quotient proposed by Ostrowsky [23]. Let A be a general (nonnormal) real square matrix of order n, and let $\mathbf{u}$ and $\mathbf{v}$ be two n-vectors that satisfy $\mathbf{u}^*\mathbf{v} \neq 0$, where $\mathbf{u}^*$ denotes the conjugate transpose of $\mathbf{u}$. Then the "generalized Rayleigh quotient"

$$\rho(\mathbf{u}, \mathbf{v}) = \mathbf{u}^* A\mathbf{v} / (\mathbf{u}^*\mathbf{v}) \tag{10.1}$$

is aimed to approximate an eigenvalue of A that is "common" to $\mathbf{u}$ and $\mathbf{v}$. One justification for this definition lies in the following observation: Let $\mathbf{u}_0$ and $\mathbf{v}_0$ be left and right eigenvectors of A corresponding to the same eigenvalue λ_0 and satisfy $\mathbf{u}_0^*\mathbf{v}_0 \neq 0$. Then $\lambda_0 = \rho(\mathbf{u}_0, \mathbf{v}_0)$. A second justification comes from the "stationary property" of $\rho(\mathbf{u}, \mathbf{v})$ at the point $(\mathbf{u}_0, \mathbf{v}_0)$, e.g., [23], [25]. These features motivate the "generalized Rayleigh quotient iteration" proposed by Ostrowsky [23], [24]. On the other hand, unlike the other quotients, it is difficult to associate (10.1) with a certain optimization problem. For further discussions of the generalized Rayleigh quotient and related topics, see [22], [23], [24], [25], [32].

The "rectangular iterations" that we propose are closely related to the power method applied to the matrices A^TA and AA^T. A similar power method iteration is used in the HITS algorithm for information retrieval, e.g., [18], [19]. However, the use of rectangular iterations has further interpretations: On one hand, it can be viewed as iterative retrieval of singular vectors, which is carried out by successive orthogonalizations. On the other hand, it is a minimization method ("point relaxation") that is aimed at solving the minimum norm problem (7.1). These observations add new insight into the power method and expose new useful results. A further merit of the minimum norm approach is that it opens the door for more sophisticated minimization techniques. The modified scheme for missing data illustrates this point. Another fruitful idea is the use of a line search acceleration; see [5], [8].

The task of computing an orthonormal basis for Range(A) is called "the orthonormal basis problem," e.g., [14]. This problem is often solved by applying Householder orthogonalization, or Gram–Schmidt orthogonalization, to produce a QR factorization of A. In practice both methods are carried out with some "column pivoting" policy, and the basis is completely determined by this policy. See, for example, [3],

[6], [7], [14], [28]. The "orthogonalization via deflation" method has larger freedom in the choice of the basis: Consider, for example, the left-side orthogonalization process. Then at the ℓth deflation stage, $\ell = 1, 2, \ldots$, the new vector that enters the basis can be any unit vector from Range(A_ℓ). The Gram–Schmidt orthogonalization process uses the ℓth column of A_ℓ, while pivoted Gram–Schmidt chooses a column of A_ℓ that has the largest Euclidean norm. The ultimate choice is $\mathbf{u}_\ell$, a left dominant singular vector of A_ℓ. However, accurate computation of this vector can be "too expensive." The theme of the paper is to point out that a few rectangular iterations with A_ℓ are likely to produce a fair estimate of $\mathbf{u}_\ell$. This way, the resulting orthogonal decomposition provides a meaningful substitute for the SVD of A. Preliminary experiments that we have done support this view; see [8].

The practical value of our approach lies in problems where standard SVD algorithms are not applicable, as in problems with missing data. Another favorable situation arises when the rank of the approximation ℓ is much smaller than $\min\{m, n\}$. In this case the algorithm performs only ℓ deflation stages, which reduces the computational cost. The last situation is likely to occur when A is a large sparse matrix. In this case $A_{\ell+1}$ is kept in the form

$$A_{\ell+1} = A - \sum_{j=1}^{\ell} \tilde{\sigma}_j \tilde{\mathbf{u}}_j \tilde{\mathbf{v}}_j^T,$$

so the matrix-vector products are able to take advantage of the sparsity pattern in A.

REFERENCES

[1] O. Alter, P. O. Brown, and D. Botstein, *Singular value decomposition for genome-wide expression data processing and modeling*, Proc. Natl. Acad. Sci. USA, 97 (2000), pp. 10101–10106.

[2] O. Alter, P. O. Brown, and D. Botstein, *Processing and modeling genome-wide expression data using singular value decomposition*, Proceedings SPIE, 4266 (2001), pp. 171–186.

[3] A. Bjorck, *Numerical Methods for Least-Squares Problems*, SIAM, Philadelphia, 1996.

[4] M. T. Chu, R. E. Funderlic, and G. H. Golub, *A rank-one reduction formula and its applications to matrix factorizations*, SIAM Rev., 37 (1995), pp. 512–530.

[5] A. Dax, *Line search acceleration of iterative methods*, Linear Algebra Appl., 130 (1990), pp. 43–63.

[6] A. Dax, *A modified Gram-Schmidt algorithm with iterative orthogonalization and column pivoting*, Linear Algebra Appl., 310 (2000), pp. 25–42.

[7] A. Dax, *Computing projections via Householder transformations and Gram-Schmidt orthogonalization*, Numer. Linear Algebra Appl., 11 (2004), pp. 675–692.

[8] A. Dax, *A Minimum Norm Approach for Low-Rank Approximations of a Matrix*, Technical report, Hydrological Service of Israel, 2006.

[9] A. Dax, *The Rectangular Quotients Equality and Related Issues*, Technical report, Hydrological Service of Israel, 2007.

[10] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[11] C. Eckhart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[12] E. Egervary, *On rank-diminishing operations and their applications to the solution of linear equations*, Z. Angew. Math. Phys., 11 (1960), pp. 376–386.

[13] S. Friedland, A. Niknejad, and L. Chihara, *A Simultaneous Reconstruction of Missing Data in DNA Microarrays*, Linear Algebra Appl., 416 (2006), pp. 8–28.

[14] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.

[15] T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, and D. Botstein, *Imputing Missing Data for Gene Expression Arrays*, Technical report, Division of Biostatistics, Stanford University, 1999.

[16] H. Hotelling, *Analysis of a complex of statistical variables into principal components*, J. Educational Psychology, 24 (1933), pp. 417–441 and 498–520.
[17] H. Hotelling, *Some new methods in matrix calculation*, Ann. Math. Stat., 14 (1943), pp. 1–34.
[18] A. N. Langville and C. D. Meyer, *The use of linear algebra by web search engines*, IMAGE, 33 (2004), pp. 2–6.
[19] A. N. Langville and C. D. Meyer, *A survey of eigenvector methods for web information retrieval*, SIAM Rev., 47 (2005), pp. 135–161.
[20] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2001.
[21] D. P. O'Leary, G. W. Stewart, and J. S. Vandergraft, *Estimating the largest eigenvalue of a positive definite matrix*, Math. Comp., 33 (1979), pp. 1289–1292.
[22] D. P. O'Leary and G. W. Stewart, *On the Convergence of a New Rayleigh Quotient Method with Applications to Large Eigenproblems*, Technical report TR-97-74, Institute for Advanced Computer Studies, University of Maryland, 1997.
[23] A. M. Ostrowski, *On the convergence of the Rayleigh quotient iteration for the computation of the characteristic roots and vectors.* III *(generalized Rayleigh quotient and characteristic roots with linear elementary divisors)*, Arch. Ration. Mech. Anal., 3 (1959), pp. 325–340.
[24] A. M. Ostrowski, *On the convergence of the Rayleigh quotient iteration for the computation of the characteristic roots and vectors.* IV *(generalized Rayleigh quotient for nonlinear elementary divisors)*, Arch. Ration. Mech. Anal., 3 (1959), pp. 341–347.
[25] B. N. Parlett, *The Rayleigh quotient iteration and some generalizations for nonnormal matrices*, Math. Comp., 28 (1974), pp. 679–693.
[26] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[27] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.
[28] G. W. Stewart, *Matrix Algorithms, Vol.* I: *Basic Decompositions*, SIAM, Philadelphia, 1998.
[29] G. W. Stewart, *Matrix Algorithms, Vol.* II: *Eigensystems*, SIAM, Philadelphia, 2001.
[30] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman, *Missing value estimation methods for DNA microarrays*, Bioinformatics, 17 (2001), pp. 520–525.
[31] J. H. M. Wedderburn, *Lectures on Matrices, Colloquium Publications*, Vol. XVII, American Mathematical Society, New York, 1934 and Dover, New York, 1964.
[32] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 261–275

THE SINKHORN–KNOPP ALGORITHM: CONVERGENCE AND APPLICATIONS*

PHILIP A. KNIGHT†

Abstract. As long as a square nonnegative matrix A contains sufficient nonzero elements, then the Sinkhorn–Knopp algorithm can be used to balance the matrix, that is, to find a diagonal scaling of A that is doubly stochastic. It is known that the convergence is linear, and an upper bound has been given for the rate of convergence for positive matrices. In this paper we give an explicit expression for the rate of convergence for fully indecomposable matrices. We describe how balancing algorithms can be used to give a measure of web page significance. We compare the measure with some well known alternatives, including PageRank. We show that, with an appropriate modification, the Sinkhorn–Knopp algorithm is a natural candidate for computing the measure on enormous data sets.

Key words. matrix balancing, Sinkhorn–Knopp algorithm, PageRank, doubly stochastic matrix

AMS subject classifications. 15A48, 15A51, 65F15, 65F35

DOI. 10.1137/060659624

1. Introduction. If a graph has the appropriate structure, we can generate a random walk on it by taking its connectivity matrix and applying a suitable scaling to transform it into a stochastic matrix. This simple idea has a wide range of applications. In particular, we can rank pages on the Internet by generating the appropriate connectivity matrix G and applying a scaling induced by a diagonal matrix D of column sums so that $P_c = GD^{-1}$ is column stochastic.[1] Ordering pages according to the size of the components in the stationary distribution of P_c gives us a ranking. Roughly speaking, this is how Google's PageRank is derived.

An alternative method of generating a random walk on G is to apply a diagonal scaling to both sides of G to form a doubly stochastic matrix $P = DGE$. Of course, if we use this approach, then the stationary distribution is absolutely useless for ranking purposes. However, in section 5 we argue that the entries of D and E can be used as alternative measures. We will also see that, if we apply the Sinkhorn–Knopp (SK) algorithm on an appropriate matrix to find D and E, we can compute our new ranking with a cost comparable to that of finding the PageRank. In order to justify this conclusion, we need to establish the rate of convergence of the SK algorithm, which we do in section 4. Before that, in section 2 we review pertinent details about the SK algorithm, and in section 3 we look at the symmetric case. Our numerical results are collected in section 6.

2. The Sinkhorn–Knopp algorithm. The SK algorithm is perhaps the simplest method for finding a doubly stochastic scaling of a nonnegative matrix A. It does this by generating a sequence of matrices whose rows and columns are normalized alternately. The algorithm can be thought of in terms of matrices

$$A_0 = A,\ A_1,\ A_2, \ldots,$$

*Received by the editors May 11, 2006; accepted for publication (in revised form) by D. Calvetti November 5, 2007; published electronically March 5, 2008. This work was supported in part by a grant from the Carnegie Trust for the Universities of Scotland.
http://www.siam.org/journals/simax/30-1/65962.html
†Department of Mathematics, University of Strathclyde, 26 Richmond Street, Glasgow G1 1XH, Scotland (pk@maths.strath.ac.uk).
[1]If any of the columns are empty, we first modify G by, for example, adding a column of ones.

whose limit is the doubly stochastic matrix we are after, or in terms of pairs of diagonal matrices

$$(D_0, E_0),\ (D_1, E_1),\ (D_2, E_2), \ldots,$$

whose limit gives the desired scaling of A. We will predominantly use the second interpretation in this paper.

To describe the algorithm more formally, we introduce the operator $\mathcal{D} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$, where $\mathcal{D}(x) = \mathrm{diag}(x)$. Starting with $D_0 = E_0 = I$, we let

$$r_k = D_{k-1} A E_{k-1} e, \tag{2.1}$$

where e is a vector of ones, and $D_k = \mathcal{D}(r_k)^{-1}$. Now let

$$c_k^T = e^T D_k A E_{k-1}, \tag{2.2}$$

and $E_k = \mathcal{D}(c_k)^{-1}$.

Not surprisingly, the simplicity of the method has led to its repeated discovery. It is claimed to have first been used in the 1930s for calculating traffic flow [5] and appeared in 1937 as a method for predicting telephone traffic distribution [15].[2] In the numerical analysis community it is most usually named after Sinkhorn and Knopp, who proved convergence results for the method in the 1960s [22], but it is also known by many other names, such as the RAS method [1] and Bregman's balancing method [16]. The letters R, A, and S represent symbolically the matrices used in decomposing the original matrix into doubly stochastic form. Many matrix decompositions are expressed in this way (e.g., CS and QT decompositions).

Perhaps the simplest representation of the method is given in [13]. Suppose that $P = \mathcal{D}(r) A \mathcal{D}(c)$ is doubly stochastic. Manipulation of the identities $Pe = e$ and $P^T e = e$ gives

$$c = \mathcal{D}(A^T r)^{-1} e, \quad r = \mathcal{D}(Ac)^{-1} e, \tag{2.3}$$

which suggests the fixed point iteration

$$c_{k+1} = \mathcal{D}(A^T r_k)^{-1} e, \quad r_{k+1} = \mathcal{D}(A c_{k+1})^{-1} e. \tag{2.4}$$

It is straightforward to show that this iteration is precisely the SK algorithm when $r_0 = e$. Note that this can be achieved by repeatedly issuing the commands

$$\mathtt{c = 1./(A' * r)}, \quad \mathtt{r = 1./(A * c)}$$

in MATLAB.

Convergence of the SK algorithm depends on the nonzero structure of A. Recall that a nonnegative matrix A has total support if $A \neq 0$ and all its nonzero elements lie on a positive diagonal. This rather terse definition is standard in the literature [19, 22] but can be usefully interpreted in terms of graphs. If a graph has an adjacency matrix with the same nonzero pattern as a matrix with total support, then every edge in the graph is part of a circuit. A matrix is fully indecomposable if it is impossible to find permutation matrices P and Q such that

$$PAQ = \begin{bmatrix} A_1 & 0 \\ A_2 & A_3 \end{bmatrix},$$

[2] A more detailed history of the method can be found in [7].

with A_1 square. This property is also known as the strong Hall property. Sinkhorn and Knopp proved the following result [22].

THEOREM 2.1 (Sinkhorn–Knopp). [illegible]

Note that we are not claiming that D and E are unique but rather that if $D_1AE_1 = D_2AE_2 = P$, then there exists $\alpha > 0$ such that $D_1 = \alpha D_2$ and $E_2 = \alpha E_1$.

By thinking of the iteration in terms of the approximate doubly stochastic matrices

$$A_0,\ A_1,\ A_2, \ldots,$$

Sinkhorn and Knopp also showed that the algorithm converges whenever A has at least one positive diagonal. For example, if we were to scale the matrix

$$\begin{bmatrix} a & b \\ 0 & c \end{bmatrix}$$

repeatedly, we would converge to the identity matrix; however, the diagonal matrices in the identity $A_k = D_kAE_k$ would diverge.

The rate of convergence of the SK algorithm has also been studied by a number of authors. Soules [23] has shown that the algorithm is linearly convergent whenever the original matrix has total support. However, he gives no explicit value for the rate of convergence. Soules establishes his result by treating the algorithm as a fixed point iteration on matrices and looking at the Jacobian matrix. Our interpretation of the method as an iteration on vectors enables us to improve this result.

Franklin and Lorenz [11] give a bound on the rate of convergence when $A > 0$. They use Hilbert's projective metric for vectors $x, y \in \mathbb{R}^n_+$, namely,

$$d(x, y) = \log \max_{i,j} \frac{x_i y_j}{x_j y_i}.$$

For $A \in \mathbb{R}^{m\times n}_+$, we can define

$$\theta(A) = \sup\{d(Ax, Ay) | x, y \in \mathbb{R}^n_+\} = \max_{i,j,k,l} \frac{a_{ik}a_{jl}}{a_{jk}a_{il}}. \tag{2.5}$$

Franklin and Lorenz show that $\theta(A) = \theta(A_m)$ is constant for the sequence of matrices $\{A_m\}$ generated by the SK algorithm with initial matrix A. They are also able to show that the rate of convergence of the method is bounded above by

$$C = \left(\frac{\sqrt{\theta(A)} - 1}{\sqrt{\theta(A)} + 1} \right)^2. \tag{2.6}$$

This is an a priori bound on the rate of convergence, but it can be very weak in practice. Furthermore, the result holds only for positive matrices. As the smallest element of A approaches zero, it can be seen that C approaches 1. The result we establish in section 4 is sharp and applies whenever A is fully indecomposable.

It is worth noting that we can generate a stopping criterion for the SK algorithm that can be computed very efficiently. We want to stop when $\mathcal{D}(r_k)Ac_k$ and $\mathcal{D}(c_k)A^T r_k$ are both close to e. After each SK step, the first of these criteria is satisfied (up to round-off error) as we will have just balanced the rows of A. To get an estimate of the error in the column sums, we note that $A^T r_k = \mathcal{D}(c_{k+1})^{-1}e$, so in the middle of the step we can estimate our error by computing

$$err_k = \|c_k \circ d_{k+1} - e\|_1, \tag{2.7}$$

where $d_{k+1} = \mathcal{D}(c_{k+1})^{-1}e$ and $\circ$ represents the Hadamard product.

Matrix balancing can be used as a simple technique for preconditioning a matrix. Given a fully indecomposable matrix $A \in \mathbb{R}^{n\times n}$ we can find two $n \times n$ diagonal matrices D and E such that the p-norms of the rows and columns of DAE are all equal. This idea was explored in [2, 10] as a method for finding a diagonal scaling such that $\kappa(DAE) \ll \kappa(A)$. By applying the SK algorithm to the matrix whose (i, j)th element is $|a_{ij}^p|$, it is easily seen that the problem is essentially identical for $1 < p < \infty$. The case $p = \infty$ is studied in [8, 20].

3. Balancing symmetric matrices. If A is symmetric, then it is natural to look for a diagonal matrix D such that DAD is doubly stochastic. We can do this by using the SK algorithm: If $\mathcal{D}(r)A\mathcal{D}(c)$ is doubly stochastic, then so is its transpose $\mathcal{D}(c)A\mathcal{D}(r)$, and since, up to a scalar factor, the balancing is unique (by Theorem 2.1), $r = \alpha c$. If $\alpha \neq 1$, we can scale our limiting vectors to regain symmetry.

During the iteration, though, symmetry is lost, and an alternative approach is to generate a sequence of symmetric iterates. The symmetric analogues of (2.3) and (2.4) are

$$x = \mathcal{D}(Ax)^{-1}e \tag{3.1}$$

and

$$x_k = \mathcal{D}(Ax_{k-1})^{-1}e \tag{3.2}$$

respectively. We note that this iteration can be coded in MATLAB by repeated application of the single instruction `x = 1./(A*x)`, which must make it one of the most compact algorithms in numerical analysis!

While the iteration superficially retains symmetry, it is in fact no different from the SK algorithm. By comparing (3.2) with (2.4) we see that, for $k \geq 0$, $x_{2k} = r_k$ and $x_{2k+1} = c_{k+1}$.

Conversely, we can use the iteration given by (3.2) on nonsymmetric matrices: Simply apply it to

$$S = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}. \tag{3.3}$$

This is more than an academic exercise. To establish the rate of convergence of the SK algorithm we first find the convergence rate of (3.2). This will be sufficient, as in exact arithmetic the iterates coincide.

To see this, let

$$x_k = \begin{bmatrix} y_k \\ z_k \end{bmatrix},$$

and (3.2) becomes

$$y_{k+1} = \mathcal{D}(Az_k)^{-1}e, \tag{3.4}$$

$$z_{k+1} = \mathcal{D}(A^T y_k)^{-1}e. \tag{3.5}$$

Hence

$$y_{k+1} = \mathcal{D}(A\mathcal{D}(A^T y_{k-1})^{-1}e)^{-1}e,$$
$$z_{k+1} = \mathcal{D}(A^T\mathcal{D}(Az_{k-1})^{-1}e)^{-1}e.$$

However, from (2.4), we have

$$c_k = \mathcal{D}(A\mathcal{D}(A^T c_{k-1})^{-1}e)^{-1}e,$$
$$r_k = \mathcal{D}(A^T\mathcal{D}(Ar_{k-1})^{-1}e)^{-1}e,$$

and we conclude that one step of the SK algorithm is equivalent to two steps of (3.2) applied to S.

Symmetric balancing is also considered in [18], where the equation $\mathcal{D}(Ax)x = e$ is solved by using a Gauss–Seidel–Newton method.

4. The rate of convergence of the Sinkhorn–Knopp algorithm. We now consider the convergence of the symmetric SK algorithm in (3.2) by adapting as necessary the standard tools for analysis of a fixed point iteration. At this stage, we restrict ourselves to fully indecomposable matrices as in this case (3.2) has a unique positive fixed point, but we will comment on the more general case (matrices with total support) at the end of the section.

There are two complications we have to consider when trying to establish convergence. The first is that, in general, the iteration does not converge as when the SK algorithm is used on a symmetric matrix the sequences $\{r_k\}$ and $\{c_k\}$ will almost surely converge to different limits. Eventually we oscillate between a pair of vectors that are scalar multiples of the fixed point. However, our ultimate goal is to establish a sharp convergence result for the general SK algorithm, and it will suffice to consider the alternating subsequences.

The second complication is that, around the fixed point, the Jacobian matrix has spectral radius one, and so we cannot make direct use of the contraction mapping theorem. However, the nature of the subspace associated with the principal eigenvector means that this, too, can be dealt with. Soules makes similar observations regarding the SK algorithm in [23] and proves linear convergence. As we are trying to put a number to this rate, we cannot use Soules's result. Instead, by using our compact representation of the iteration, we present a simple analysis that leads to an explicit value for the rate of convergence.

We first prove a couple of lemmas to confirm some of the statements made in the preceding discussion.

LEMMA 4.1. [illegible] A [illegible] x_* [illegible] $\mathcal{D}(x_*)A\mathcal{D}(x_*) = P$ [illegible] P [illegible].

. This is a trivial consequence of Theorem 2.1. For existence, suppose that $\mathcal{D}(r)A\mathcal{D}(c) = P$, and let $x_* = \sqrt{\mathcal{D}(r)\mathcal{D}(c)}e$ (by symmetry r and c are collinear). If $\mathcal{D}(x)A\mathcal{D}(x) = \mathcal{D}(y)A\mathcal{D}(y)$, then, for some $\alpha > 0$, $x = \alpha y$ and $y = \alpha x$. Hence $x = y$. □

LEMMA 4.2. [illegible] A [illegible] x_* [illegible] $\mathcal{D}(x_*)A\mathcal{D}(x_*) = P$ [illegible].

P [illegible] $f(x) = \mathcal{D}(Ax)^{-1}e$ [illegible] $f(x)$ [illegible]

1. [illegible] $x \in \mathbb{R}^n_+$, $J(x) = -\mathcal{D}(Ax)^{-2}A$
2. [illegible] $\alpha \in \mathbb{R}_+$,

$$J(\alpha x_*) = -\frac{1}{\alpha^2}\mathcal{D}(x_*)P\mathcal{D}(x_*)^{-1}.$$

[illegible]

1. This can be confirmed by a straightforward componentwise calculation or by tensor calculus. We restrict ourselves to positive vectors to ensure that $Ax > 0$ and hence that $\mathcal{D}(Ax)$ is invertible.

2. At the fixed point, $\mathcal{D}(Ax_*) = \mathcal{D}(x_*)^{-1}$; hence $\mathcal{D}(A(\alpha x_*)) = \alpha\mathcal{D}(x_*)^{-1}$ and

$$J(\alpha x_*) = -\frac{1}{\alpha}^2\mathcal{D}(x_*)^2A = \mathcal{D}(x_*)(\mathcal{D}(x_*)A\mathcal{D}(x_*))\mathcal{D}(x)^{-1} = -\frac{1}{\alpha}^2\mathcal{D}(x_*)P\mathcal{D}(x_*)^{-1}. \quad \square$$

We now consider the behavior of $f(x)$ when x is in the neighborhood of αx_*. Because of the alternating behavior, we consider the effects of two iterations at a time.

LEMMA 4.3. [illegible] A [illegible] $\mathcal{D}(x_*)A\mathcal{D}(x_*) = P$ [illegible] x_* [illegible] P [illegible] $f(x) = \mathcal{D}(Ax)^{-1}e$ [illegible] $\alpha > 0$ [illegible] $\widehat{x}$ [illegible] ϵ [illegible] αx_* [illegible]

$$\min_{v\in\mathcal{V}} \|f^2(\widehat{x}) - v\| \le |\lambda_2|^2\epsilon + o(\epsilon), \tag{4.1}$$

[illegible] $\mathcal{V}$ [illegible] x_* [illegible]. Suppose that, for some $\epsilon > 0$, $\widehat{x} = \alpha x_* + d$, with $\|d\| < \epsilon$. Let $D = \mathcal{D}(x_*)$, and note that $f(\alpha x_*) = x_*/\alpha$ and $f^2(\alpha x_*) = \alpha x_*$. We can write

$$\begin{aligned} f^2(\widehat{x}) &= f(f(\alpha x_*) + J(\alpha x_*)d + o(\epsilon)) \\ &= f^2(\alpha x_*) + J(x_*/\alpha)J(\alpha x_*)d + o(\epsilon) \\ &= \alpha x_* + (-\alpha^2 DPD^{-1})(-\alpha^{-2}DPD^{-1})d + o(\epsilon) \\ &= \alpha x_* + DP^2D^{-1}d + o(\epsilon) = \alpha x_* + J^2d + o(\epsilon), \end{aligned}$$

where $J = DPD^{-1}$. As $\rho(P) = 1$, we cannot use the contraction mapping theorem to show that $\|f^2(\widehat{x}) - \alpha x_*\| < \|\widehat{x} - \alpha x_*\|$. However, observe that A is fully indecomposable; hence P is, too, and since doubly stochastic matrices with this property are primitive [3], P has a single simple eigenvalue of modulus one. The corresponding eigenvector of J is x_*. By using Wielandt deflation [26], we can write

$$J = -(x_*y^T + J_0),$$

where

$$\sigma(J_0) = \sigma(P) - \{1\} \cup \{0\} = \{\lambda_2, \ldots, \lambda_n, 0\}$$

by choosing, for example, $y = x_*/{x_*}^T x_*$. Since $J_0x_* = 0$,

$$\begin{aligned} f^2(\widehat{x}) &= \alpha x_* + (x_*y^T + J_0)^2 d + o(\epsilon) \\ &= J_0^2 d + (1 + y^T(J_0 + I)d)x_* + o(\epsilon). \end{aligned}$$

Choosing our norm so that $\|J_0\| \leq |\lambda_2| + \epsilon$ and letting $v = (1 + y^T(J_0 + I)d)x_*$ establishes (4.1). □

We can conclude that, as our iterates approach the subspace spanned by x_*, the contribution to our iterates from other directions diminishes linearly at a rate governed by the second eigenvalue of P. The fact that we are heading for a fixed line rather than a fixed point is sufficient for us to find the scaling we crave. Since we already know that the SK algorithm converges, we can be sure that we eventually lie in a neighborhood that satisfies the conditions of Lemma 4.3.

THEOREM 4.4. [illegible] A [illegible] $\mathcal{D}(x_*)A\mathcal{D}(x_*) = P$ [illegible] P [illegible] $\{x_k\}$ [illegible] (3.2) [illegible] $x_0 = e$ [illegible] $\epsilon > 0$ [illegible] $K_1 \in \mathbb{Z}$ [illegible] $k \geq K_1$, $x_k = \alpha_k x_* + d_k$ [illegible] $\|d_k\| < \epsilon$ [illegible] α_k [illegible] $K_2 \in \mathbb{Z}$ [illegible] $k \geq K_2$,

$$\|d_{k+2}\| \leq |\lambda_2|^2 \|d_k\|,$$

[illegible] λ_2 [illegible] P [illegible]. The existence of K_1 is guaranteed by Theorem 2.1 and our observation on the equivalence of the SK algorithm and (3.2). The existence of K_2 follows from Lemma 4.3. □

The result does not immediately extend to the nonsymmetric case as when we form S by using (3.3) we lose indecomposability. This isn't a problem though.

THEOREM 4.5. [illegible] A [illegible] r_* [illegible] c_* [illegible] $\mathcal{D}(r_*)A\mathcal{D}(c_*) = P$ [illegible] P [illegible] $K \in \mathbb{Z}$ [illegible] $k \geq K$,

$$\left\| \begin{bmatrix} r_{k+1} \\ c_{k+1} \end{bmatrix} - \begin{bmatrix} r_* \\ c_* \end{bmatrix} \right\| \leq \sigma_2^2 \left\| \begin{bmatrix} r_k \\ c_k \end{bmatrix} - \begin{bmatrix} r_* \\ c_* \end{bmatrix} \right\|,$$

[illegible] σ_2 [illegible] P [illegible]. The convergence of the algorithm is guaranteed by Theorem 2.1. To determine the rate of convergence we need to adapt Lemma 4.3. Consider the spectrum of $J(x_*)$ when we form the matrix S by using (3.3). This will be the same as the spectrum of

$$Q = \begin{bmatrix} 0 & P \\ P^T & 0 \end{bmatrix}.$$

The conditions imposed on A ensure that P is primitive, and hence so is P^TP. Since the spectrum of Q is the set of positive and negative square roots of the eigenvalues[3] of P^TP, we have an additional eigenvalue of modulus one. We need to consider how the iteration behaves in the neighborhood of the associated subspace $\mathcal{V}$.

The two eigenvectors of $J(x_*)$ corresponding to the maximal eigenvalues take the form

$$v_1 = \begin{bmatrix} r_* \\ c_* \end{bmatrix} \quad \text{and} \quad v_2 = \begin{bmatrix} r_* \\ -c_* \end{bmatrix}.$$

By assuming $\widehat{x}$ is in an ϵ-neighborhood of $\mathcal{V}$, we can again show that

$$\min_{v \in \mathcal{V}} \|f^2(\widehat{x}) - v\| \leq |\lambda_2(Q)|^2 \epsilon + o(\epsilon),$$

and $|\lambda_2(Q)| = \sigma_2(P)$.

[3] Or singular values of P.

We have essentially proved the theorem; we just have to identify the iterates from the symmetric algorithm that appear as iterates in the SK algorithm. By following the discussion at the end of section 3 we can identify r_k as the top half of x_{2k} and c_k as the bottom half of x_{2k-1}. This explains why the rate of convergence of the SK algorithm is σ_2^2. SK algorithm avoids the oscillations in the symmetric algorithm as r_k and c_k are formed from convergent subsequences of $\{x_k\}$. □

Theorem 2.1 states that the SK algorithm is convergent if A has total support, while Theorem 4.5 applies only if A is fully indecomposable. This gap is easily reconciled: If A has total support but is not fully indecomposable, then it must be a direct sum of fully indecomposable matrices. Such a matrix can be permuted into block diagonal form

$$\begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_k \end{bmatrix},$$

where each diagonal block is fully indecomposable. The behavior of the SK algorithm is unaffected by permutations (unlike the symmetric variant). If we apply the SK algorithm to the block diagonal form, then clearly the convergence in each block will be independent of all others, and the doubly stochastic matrix we converge towards can be written

$$\begin{bmatrix} P_1 & & & \\ & P_2 & & \\ & & \ddots & \\ & & & P_k \end{bmatrix},$$

where each P_i is itself doubly stochastic and fully indecomposable. The asymptotic rate of convergence to P_i is $\sigma_2^2(P_i)$. If we want to talk about an overall asymptotic convergence, then it will be bounded above by

$$\max_{1 \le i \le k} \sigma_2^2(P_i).$$

However, we may not see this upper bound reached, for example, in the case that some of the A_i are already doubly stochastic.

5. Ranking web pages. The PageRank algorithm, introduced by Brin et al. [6], has proved to be an incredibly successful technique for ordering large sets of connected data. In essence, the method takes a matrix G representing the connectivity of a network and scales the columns so that the matrix is column stochastic.[4] The stationary distribution of this scaled matrix is then calculated, typically by using the power method, and the size of the probabilities is used to order the nodes in the network. A thorough description of the method and associated theory can be found in [17]. We note that the column scaling is trivial to achieve (requiring half a step of the SK algorithm) and the main work is in computing the stationary distribution. In this section we use the SK algorithm to compute an alternative method for ordering data which has a similar cost to PageRank but which has two principal advantages. First,

[4] In our connectivity matrix the (i, j)th entry is one if there is a link from the jth node to the ith node.

for each node in our network we get two measures rather than one which we claim are analogous to the authorities and hubs of Kleinberg's Hypertext Induced Topic Selection (HITS) algorithm [14]. Second, there is no need to treat dangling nodes differently to any other, whereas in the PageRank algorithm it is necessary to preprocess the connectivity matrix in some way, because otherwise the column scaling fails [21].

The guiding heuristic behind the PageRank model is simple to state, namely, that the random walk will visit significant web pages more frequently than insignificant ones, and the success of this graph interpretation in mimicking the subjective property of significance is one of the main reasons behind its current ubiquity.

We offer a simple heuristic to justify our application of the SK algorithm to the problem. Clearly the probabilities in the associated distribution tell us nothing as the distribution is uniform.[5] If we think in terms of the traffic flowing around the network represented by G, then our aim is to balance the flow through each node. That is, we want to scale G so that its stationary distribution is uniform or, equivalently, so that it is doubly stochastic. Suppose then that $\mathcal{D}(r)G\mathcal{D}(c)$ is doubly stochastic. If node i in the unweighted graph draws traffic in disproportionately, then this will have to be compensated for by r_i being relatively small. Similarly, if a node has a tendency to emit traffic, then c_i will need to be relatively small. We associate the tendency of a node to emit traffic with it being a hub, a node which points to several sources of information on a topic. The tendency to draw in traffic is associated with authoritativeness, a node that contains definitive information on a particular topic. We can order the nodes with respect to each of these properties by reversing the order of the entries of r and c. This heuristic is very similar to that behind the ordinary gravity model in transport planning [1, 16], where the SK algorithm has been successfully employed.

While we believe the use of the SK algorithm in web applications is new, it is related to a technique proposed by Tomlin in [24]. Here one looks to find a vector d such that similarity transformation induced by $\mathcal{D}(d)$ on the connectivity matrix $P = D^{-1}GD$ fixes the sum of the entries of P and, for $1 \le i \le n$,

$$\sum_{j=1}^{n}(p_{ij} - p_{ji}) = 0. \tag{5.1}$$

Tomlin argues that the authoritativeness of the jth node is proportional to the size of d_j, while the jth row/column sum can be used as a hub measure. Tomlin suggests an iterative algorithm for computing d, the iterative step for which can be written in MATLAB as

$$\texttt{d} = \texttt{sqrt}((\texttt{G} * \texttt{d})./(\texttt{G}' * (1./\texttt{d}))),$$

but no conditions for convergence are given although it is claimed to work in practice. A criticism of Tomlin's technique is that, if G is symmetric, (5.1) is satisfied with $D = I$, and the method fails to identify authorities. While G will not be symmetric in web applications, there seems to be no justification for this phenomenon.

5.1. Practicalities. On any large set of web data it is unreasonable to expect the nodes to form a single strongly connected component, and our matrix is highly unlikely to be fully indecomposable. Hence it is necessary to make a perturbation to G for the SK algorithm to converge. In PageRank a damping factor is used: If P

[5] We can claim categorically that this is the worst possible method for ranking web pages!

```
function [c, r] = sk(G, tol, g)
[n, n] = size(G);
r = ones(n,1); c = r;
d = G'*r + g*sum(r);
while norm(c.*d - 1,1) > tol
    c = 1./d;
    r = 1./(G*c+ g*sum(c));
    d = G'*r + g*sum(r);
end
```

FIG. 5.1. *A balancing algorithm for web ranking.*

is the column stochastic scaling of the web graph, then we compute the stationary distribution of

$$P_\alpha = \alpha P + (1-\alpha)ee^T/n. \tag{5.2}$$

Inspired by this idea, we simply make a rank one perturbation to G by adding a constant γ to each element. Our justification for doing this is similar to that in PageRank: If we wish to model a random crawl on the web, we have to allow a mechanism for moving between any pair of nodes. Clearly we do not want to make the perturbation explicitly as we want to take advantage of the sparsity in G, and indeed it is easily avoided. By using (2.4), and the fact that all of the iterates are positive, we can write

$$c_{k+1} = \mathcal{D}((G+\gamma ee^T)^T r_k)^{-1} e = \mathcal{D}(G^T r_k + \gamma \|r_k\|_1 e)^{-1} e$$

and similarly

$$r_{k+1} = \mathcal{D}(G^T c_{k+1} + \gamma \|c_{k+1}\|_1 e)^{-1} e.$$

A MATLAB program for carrying out balancing of $G + \alpha ee^T$ by using the stopping criterion for the SK algorithm (2.7) is given in Figure 5.1. All the user needs to supply is the connectivity graph and a choice of tolerance and the parameter γ. The cost of the algorithm is dominated by the two matrix-vector multiplies at each step. For very large values of n, the cost of the transpose is likely to be significant, and the algorithm should be adapted to work with G and G^T efficiently.

The damping factor in PageRank controls the rate of convergence of the power method by fixing the size of the second eigenvalue of P_α. This is a consequence of the following theorem, due to Brauer [4], a simple proof of which can be found in [9].

THEOREM 5.1. [illegible]

$$1, \lambda_2, \ldots, \lambda_n.$$

[illegible] $0 \le \alpha \le 1$ [illegible] P_α [illegible] (5.2) [illegible] $\{1, \alpha\lambda_2, \ldots, \alpha\lambda_n\}$

The result is also true for a more general set of rank one perturbations, but if we restrict ourselves to this particular one, we can extend the result to determine the singular values in the doubly stochastic case.

COROLLARY 5.2. [illegible]

$$1, \sigma_2, \ldots, \sigma_n.$$

[illegible] $0 \le \alpha \le 1$ [illegible] P_α [illegible] $\{1, \alpha\sigma_2, \ldots, \alpha\sigma_n\}$

. Since

$$\begin{aligned} P_\alpha^T P_\alpha &= \alpha^2 P^T P + \frac{\alpha(1-\alpha)}{n}(ee^T P + Pee^T) + \frac{(1-\alpha)^2}{n^2} ee^T ee^T \\ &= \alpha^2 P^T P + \frac{2\alpha(1-\alpha)}{n} ee^T + \frac{(1-\alpha)^2}{n} ee^T \\ &= \alpha^2 P^T P + \frac{1-\alpha^2}{n} ee^T, \end{aligned}$$

the result follows by applying Theorem 5.1 to $P^T P$. □

In many applications, α is given the value 0.85, but care must be taken to ensure that P_α sufficiently resembles P [12]. For the balancing algorithm we are unable to prove a result as strong as Theorem 5.1. However, by using our convergence result for the SK algorithm, we argue that the criteria for making a good choice for the parameter γ are similar to those used in PageRank.

We can apply the Franklin–Lorenz bound (2.6) in the perturbed case to get an idea of the effect of varying γ. Since G contains only zeros and ones we have, from (2.5),

$$\theta(G + \gamma ee^T) = \max_{i,j,k,l} \frac{(g_{ik}+\gamma)(g_{jl}+\gamma)}{(g_{jk}+\gamma)(g_{il}+\gamma)} \leq \frac{(1+\gamma)^2}{\gamma^2},$$

and hence the rate of convergence can be bounded above by $1/(1+2g)$. While this shows that we can expect the convergence of the algorithm to improve by increasing γ, experimental evidence shows that this severely underestimates the effect of the parameter, and a more realistic upper bound would be of the form $1/p(n,\gamma)$ for some low degree polynomial in n and γ. Such a bound is simple to prove in certain important special cases.

For example, suppose that P is doubly stochastic, and we use the SK algorithm on $P' = P + \gamma ee^T$. Then $\mathcal{D}(r_k)P'\mathcal{D}(c_k)$ converges to $Q = (1+n\gamma)^{-1}P'$, since this is clearly a doubly stochastic diagonal scaling of P', and, by Theorem 2.1, such a scaling is unique. Notice that $Q = P_\alpha$, where $\alpha = (1+n\gamma)^{-1}$, and so, by Corollary 5.2 and Theorem 4.5, the SK algorithm will converge asymptotically with rate $(1+n\gamma)^{-2}$. For example, choosing $\gamma = 0.1/n$ gives a convergence rate of around 0.83.

6. Results. In section 4 we showed that if the SK algorithm is used on a fully indecomposable nonnegative matrix and it converges to the doubly stochastic matrix P, then the rate of convergence is asymptotically equal to the square of the second singular value of P. Generally, we have found that this asymptotic convergence rate is approached fairly quickly. This is illustrated in Figure 6.1 for three matrices. A is the 10×10 upper Hessenberg matrix whose nonzero entries are all 1; B and C are random 50×50 matrices whose nonzero entries are uniformly distributed in $[0,1]$. They are generated so that approximately 30 percent of B's elements and 15 percent of C's elements are nonzero. The solid lines show the error as the iteration progresses using (2.7); the dashed lines represent the asymptotic rates predicted by Theorem 4.5.

In section 5.1 we claimed that the rate of convergence of the SK algorithm was significantly faster when we made a uniform rank one perturbation to the original graph. In Figures 6.2 and 6.3 we provide evidence for our claim that the rate of convergence of the SK algorithm on the $n \times n$ matrix $A + \gamma ee^T$ can be bounded by $1/p(n,\gamma)$ for some low degree polynomial in n and γ.

In Figure 6.2 we show the results of varying γ on a sparse random symmetric 1000×1000 matrix with a positive diagonal (which ensures that the matrix is fully indecomposable).

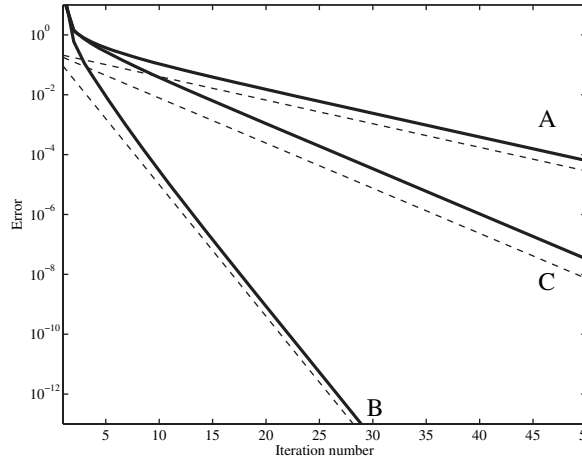


FIG. 6.1. *Rate of convergence of the Sinkhorn–Knopp algorithm.*

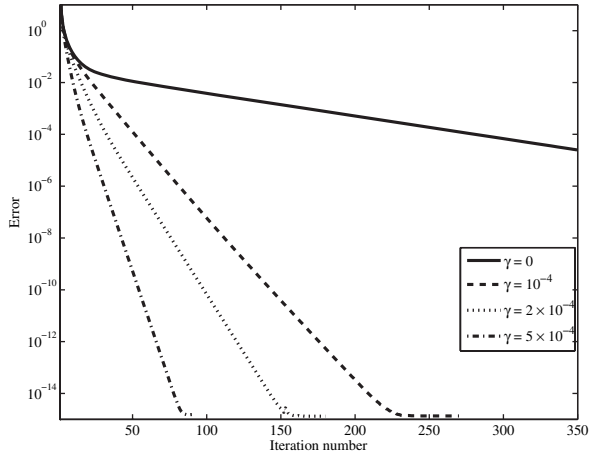


FIG. 6.2. *Varying γ for a random sparse matrix.*

In Figure 6.3 we show the results of varying γ on the connectivity graph for a 2002 web crawl of Stanford University websites [27]. There are 281093 nodes, and the matrix has roughly 2 million nonzero entries. While this particular matrix has been criticized [25] for not having a representative web structure, it usefully illustrates the effects of varying γ. In this case, if $\gamma = 0$, the matrix is not fully indecomposable. The lines show how convergence speeds up as we vary γ through the values $0.01/n$, $0.1/n$, $0.5/n$, $1/n$, $2/n$, and $4/n$.

We now investigate how our new measure compares with PageRank. In our first example, we look at the toy example of a graph of six web pages used in [17], whose connectivity is illustrated in Figure 6.4.

By using PageRank with $\alpha = .9$, the nodes are ordered (from most significant to least) 4, 6, 5, 2, 3, 1. By using the HITS algorithm, the order of authoritativeness is 5, 2, 6, 1, 4, 3, while the hub ordering is 3, 4, 1, 5, 6, 2. By using the algorithm in Figure 5.1 (and with $\gamma = 1/60$) we find that our ordering of authoritativeness matches PageRank exactly. Our ordering of the hubs differs from HITS only in that nodes

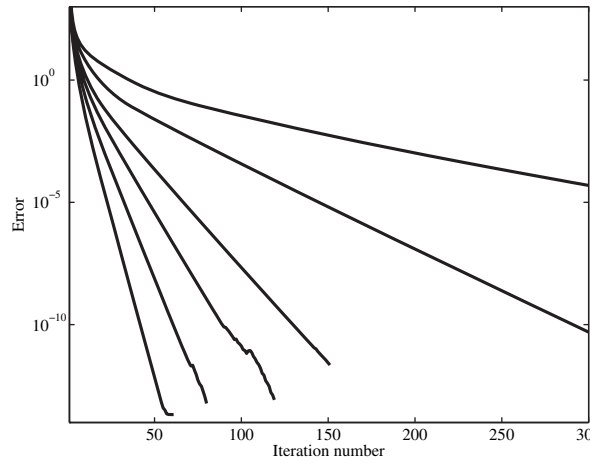


FIG. 6.3. *Varying γ for the Stanford matrix.*

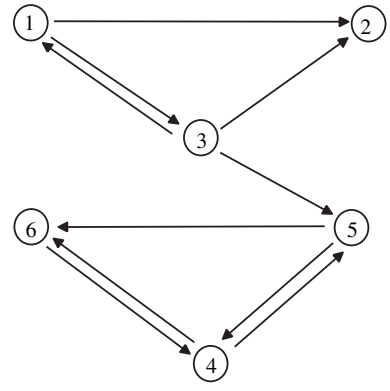


FIG. 6.4. *A miniature web graph.*

1 and 4 are transposed. We should not expect the exact correspondence between PageRank and our new measure to extend to larger systems as we are trying to measure something different.

We have carried out a number of experiments on the graph of all of the links between articles in the Wikipedia online database, collated in 2005. The resulting graph has just over 1.1 million nodes, and there are roughly 18.3 million nonzeros in the connectivity matrix. Figure 6.5 shows a comparison of PageRank ($\alpha = .85$) against the authorities computed by the SK algorithm ($\gamma = .1/n$). The graph shows the proportion of nodes that are among the top N authorities and are in the top N for high PageRank for $1 \leq N \leq 1000$. We note the strong correlation between the two.

Finally, we investigate how well the SK algorithm allows us to distinguish between hubs and authorities. Table 6.1 shows the top 10 or so nodes[6] in the Wikipedia dataset according to a variety of measures. The first column is ordered according to PageRank ($\alpha = .85$) and the second according to the authorities as measured with the SK algorithm. In the third column we have filtered out authorities whose hub rating

[6]We have grouped certain linked terms that appeared consecutively.

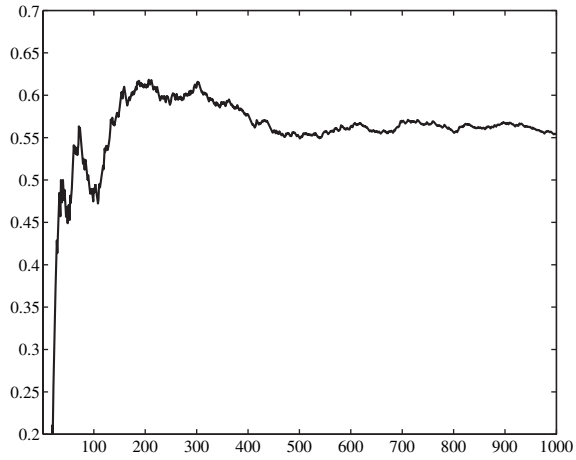


FIG. 6.5. *Comparison of web measures on Wikipedia data.*

TABLE 6.1
Highest ranked subjects in Wikipedia.

PageRank	Authorities	Filtered auth.	Filtered hubs
United States	2000	2000	Political parties
Race (U.S. Census)	Pop. density	Marriage	Environment topics
United Kingdom	km^2	U.S.	State leaders
France	Census	2003,2004,2005	Airlines
2005,2004,2000	Square mile	UK/England	2 letter combinations
Canada	Marriage	Canada	Masts
England	Per capita income	Japan	Mathematicians
Cat. by country	U.S. Census	Australia	Peerage of the UK
2003	Poverty line	2001, 2002	Record labels
Cat.:Culture	Race (U.S. Census)	Germany	Biblical names

is particularly low. Our rationale for doing this is that if an authoritative page has a high hub rating it will be linked to many other subjects and is therefore likely to be of more general interest. This is precisely what we see here, where we have listed only authorities that are also in the top 2% of hubs. The fourth column lists the top hubs, this time filtered to include only those among the top 25% of authorities. We note that all of the top hubs are either tables or lists.

7. Concluding remarks. The SK algorithm can be viewed (for symmetric matrices) as a power method like technique for solving the matrix problem $Ax = 1/x$. This connection can be seen in the similar convergence properties and costs of the two algorithms. The results of our experiments back our claim that the SK algorithm can be used to distinguish between hubs and authorities in web-type graphs at a cost similar to that of PageRank. The notion of quality of an ordering is fairly subjective, but we feel that the results in Table 6.1 demonstrate that we can obtain useful information with this approach.

In order to balance speed and quality in ordering web data with the algorithm given in Figure 5.1, we suggest choosing the parameter γ to lie in the range $.01 \leq \gamma n \leq 1$. Evidence that a choice in this range can be used to compute a measure in a comparable time to PageRank is supplied by our experiments and the partial results in section 5.1.

Acknowledgment. We thank David Gleich at Stanford University for providing the data from Wikipedia.

REFERENCES

[1] M. Bacharach, *Biproportional Matrices & Input-Output Change*, Cambridge University Press, London, 1970.
[2] F. L. Bauer, *Optimally scaled matrices*, Numer. Math., 5 (1963), pp. 73–87.
[3] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1994.
[4] A. Brauer, *Limits for the characteristic roots of a matrix.* IV: *Applications to stochastic matrices*, Duke Math. J., 19 (1952), pp. 75–91.
[5] L. M. Bregman, *Proof of the convergence of Sheleikhovskii's method for a problem with transportation constraints*, Comput. Math. Math. Phys., 1 (1967), pp. 191–204.
[6] S. Brin, L. Page, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Technical report 1999-0120, Computer Science Department, Stanford University, 1999.
[7] J. B. Brown, P. J. Chase, and A. O. Pittenger, *Order independence and factor convergence in iterative scaling*, Linear Algebra Appl., 190 (1993), pp. 1–38.
[8] J. R. Bunch, *Equilibration of symmetric matrices in the max-norm*, J. Assoc. Comput. Mach., 18 (1971), pp. 566–572.
[9] L. Eldén, *A Note on the Eigenvalues of the Google Matrix*, Technical report LiTH-MAT-R-04-01, Department of Mathematics, Linköping University, 2004.
[10] G. E. Forsythe and E. G. Straus, *On best conditioned matrices*, Proc. Amer. Math. Soc., 6 (1955), pp. 340–355.
[11] J. Franklin and J. Lorenz, *On the scaling of multidimensional matrices*, Linear Algebra Appl., 114/115 (1989), pp. 717–735.
[12] G. H. Golub and C. Grief, *An Arnoldi-type algorithm for computing PageRank*, BIT, 46 (2006), pp. 759–771.
[13] B. Kalantari and L. Khachiyan, *On the complexity of nonnegative-matrix scaling*, Linear Algebra Appl., 240 (1996), pp. 87–103.
[14] J. M. Kleinberg, *Authoritative sources in a hyperlinked environment*, J. Assoc. Comput. Mach., 46 (1999), pp. 604–632.
[15] R. Kruithof, *Telefoonverkeersrekening*, De Ingenieur, 52 (1937), pp. E15–E25.
[16] B. Lamond and N. F. Stewart, *Bregman's balancing method*, Transportation Research, 15B (1981), pp. 239–248.
[17] A. N. Langville and C. S. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ, 2006.
[18] O. E. Livne and G. H. Golub, *Scaling by binormalization*, Numer. Algorithms, 35 (2004), pp. 97–120.
[19] B. N. Parlett and T. L. Landis, *Methods for scaling to doubly stochastic form*, Linear Algebra Appl., 48 (1982), pp. 53–79.
[20] D. Ruiz, *A Scaling Algorithm to Equilibrate Both Rows and Columns Norms in Matrices*, Technical report RT/APO/01/4, ENSEEIHT-IRIT, 2001.
[21] S. Serra Capizzano, *Google pageranking problem: The model and the analysis*, in Dagstuhl Seminar Proceedings 07071: Information Retrieval and Linear Algebra Algorithms, A. Frommer, M. W. Mahoney, and D. B. Szyld, eds., 2007.
[22] R. Sinkhorn and P. Knopp, *Concerning nonnegative matrices and doubly stochastic matrices*, Pacific J. Math. 21 (1967), pp. 343–348.
[23] G. W. Soules, *The rate of convergence of Sinkhorn balancing*, Linear Algebra Appl., 150 (1991), pp. 3–40.
[24] J. A. Tomlin, *A new paradigm for ranking pages on the World Wide Web*, in Proceedings of the 12th International Conference on World Wide Web, 2003, International World Wide Web Conference Committee, pp. 350–355.
[25] S. Vigna, *Stanford matrix considered harmful*, in Dagstuhl Seminar Proceedings 07071: Information Retrieval and Linear Algebra Algorithms, A. Frommer, M. W. Mahoney, and D. B. Szyld, eds., 2007.
[26] H. Wiedlandt, *Das Iterationsverfahren bei nicht selbstadjungierten linearen Eigenwertaufgaben*, Math. Z., 50 (1944), pp. 93–143.
[27] Available from http://www.stanford.edu/∼sdkamvar/data/stanford-web.tar.gz.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 276–290

A FAST NEWTON'S METHOD FOR A NONSYMMETRIC ALGEBRAIC RICCATI EQUATION*

DARIO A. BINI†, BRUNO IANNAZZO‡, AND FEDERICO POLONI§

Abstract. A special instance of the algebraic Riccati equation $XCX - XE - AX + B = 0$ where the $n \times n$ matrix coefficients A, B, C, E are rank structured matrices is considered. Relying on the structural properties of Cauchy-like matrices, an algorithm is designed for performing the customary Newton iteration in $O(n^2)$ arithmetic operations (ops). The same technique is used to reduce the cost of the algorithm proposed by L.-Z. Lu in [*Numer. Linear Algebra Appl.*, 12 (2005), pp. 191–200] from $O(n^3)$ to $O(n^2)$ ops while still preserving quadratic convergence in the generic case. As a byproduct we show that the latter algorithm is closely related to the customary Newton method by simple formal relations. In critical cases where the Jacobian at the required solution is singular and quadratic convergence turns to linear, we provide an adaptation of the shift technique in order to get rid of the singularity. The original equation is transformed into an equivalent Riccati equation where the singularity is removed while the matrix coefficients maintain the same structure as in the original equation. This leads to a quadratically convergent algorithm with complexity $O(n^2)$ which provides approximations with full precision. Numerical experiments and comparisons which confirm the effectiveness of the new approach are reported.

Key words. nonsymmetric algebraic Riccati equation, Newton's iteration, Cauchy matrix, matrix equation, fast algorithm, M-matrix

AMS subject classifications. 15A24, 65F05, 65H10

DOI. 10.1137/070681478

1. Introduction.

Consider the following nonsymmetric algebraic Riccati equation (NARE) arising in transport theory:

$$XCX - XE - AX + B = 0, \tag{1.1}$$

where $A, B, C, E \in \mathbb{R}^{n\times n}$ are given by

$$A = \Delta - eq^T, \quad B = ee^T, \quad C = qq^T, \quad E = D - qe^T, \tag{1.2}$$

and

$$\begin{array}{lll} e = (1, 1, \ldots, 1)^T, & & \\ q = (q_1, q_2, \ldots, q_n)^T & \text{with} & q_i = \frac{w_i}{2t_i}, \\ \Delta = \operatorname{diag}(\delta_1, \delta_2, \ldots, \delta_n) & \text{with} & \delta_i = \frac{1}{ct_i(1+\alpha)}, \\ D = \operatorname{diag}(d_1, d_2, \ldots, d_n) & \text{with} & d_i = \frac{1}{ct_i(1-\alpha)}. \end{array} \tag{1.3}$$

The matrices and vectors above depend on the parameters $0 < c \leqslant 1$, $0 \leqslant \alpha < 1$ and on the sequences $0 < t_n < \cdots < t_2 < t_1 < 1$ and $w_i > 0$, $i = 1, 2, \ldots, n$, such

*Received by the editors January 31, 2007; accepted for publication (in revised form) by J. H. Brandts November 16, 2007; published electronically March 5, 2008. This work was partially supported by MIUR grant 2006017542.

http://www.siam.org/journals/simax/30-1/68147.html

†Dipartimento di Matematica, Università di Pisa, Largo B. Pontecorvo 5, 56127 Pisa, Italy (bini@mail.dm.unipi.it).

‡Dipartimento di Matematica, Università di Pisa, Largo B. Pontecorvo 5, 56127 Pisa, Italy. Current address: Dipartimento di Fisica e Matematica, Università dell'Insubria, Via Valleggio, 11, Como I-22100, Italy (bruno.iannazzo@uninsubria.it).

§Scuola Normale Superiore, Piazza dei Cavalieri 6, 56126 Pisa, Italy (poloni@sns.it).

that $\sum_i w_i = 1$. For more details and for the physical meaning of these parameters, we refer the reader to [13] and to the references therein. The solution of interest is the minimal positive one, which exists as proved by Juang and Lin in [13].

It was shown by Guo [6] that this equation falls in the class of NAREs associated with a nonsingular M-matrix or a singular irreducible M-matrix; in fact, arranging the coefficients as

$$M = \begin{bmatrix} E & -C \\ -B & A \end{bmatrix} \tag{1.4}$$

yields an M-matrix.

For this class of AREs, several suitable algorithms exist for computing the minimal positive solution: the Newton method [10], the logarithmic and cyclic reduction [2, 7], and the structure-preserving doubling algorithm [9, 11]. All these algorithms share the same order of complexity, that is, $O(n^3)$ arithmetic ops per step, and provide quadratic convergence in the generic case.

Observing that (1.1) is defined by a linear number of parameters, it is quite natural to aim to design algorithms which exploit the structure of the matrices and thus have a cost of order lower than $O(n^3)$ ops.

A step in this direction has been done by Lu [15] who has designed a vector iteration whose limit allows one to easily recover the solution. The iteration has a computational cost of $O(n^2)$ ops per step and converges linearly for $\alpha \neq 0$ or $c \neq 1$. The linear convergence is a drawback since the algorithm in many cases needs a large number of iterations to [illegible] and it is outperformed by algorithms with quadratic convergence and $O(n^3)$ ops. In fact, the same author in [14] proposes a mixed algorithm to speed up the computation. The algorithm starts with the linear iteration of complexity $O(n^2)$ and switches to a quadratically convergent one, of complexity $O(n^3)$, when some conditions are satisfied.

In this paper we consider the customary Newton method applied to (1.1). By exploiting the rank structure of the matrix coefficients, we design an algorithm for performing the Newton step in $O(n^2)$ ops. The new approach relies on a suitable modification of the fast LU factorization algorithm for Cauchy-like matrices proposed by Gohberg, Kailath, and Olshevsky in [4].

The same idea is applied to implement the quadratically convergent iteration of Lu [14] by an algorithm with cost $O(n^2)$ ops. We also provide formal relations between the sequences generated by Lu's and Newton's iterations which enable one to deduce the convergence of Lu's algorithm directly from the well-known properties of Newton's method.

In the critical but still important case where the Jacobian at the solution is singular, the convergence of Newton's (and therefore Lu's) iteration turns to linear; also the mixed iteration proposed in [14] loses its quadratic convergence while the iteration of [15] converges sublinearly.

In this case, which is encountered when $\alpha = 0$, $c = 1$, we can get rid of the singularity of the Jacobian and consequently all the above-mentioned drawbacks. The idea is to apply the shift technique originally introduced by He, Meini, and Rhee [12] and used in the framework of Riccati equations by Guo, Iannazzo, and Meini in [9] and by Guo [7]. With this technique, we replace the original Riccati equation with a new one having the same minimal solution as the original equation (1.1) but where the singularity of the Jacobian is removed. We prove that the matrix coefficients of the new equation share the same rank structure properties of the coefficients of (1.1). This enables us to design a fast Newton iteration which preserves the quadratic

convergence and keeps the same $O(n^2)$ complexity even in the critical case.

As a byproduct of this analysis, we find that the approximation to the minimal solution of (1.1) that we compute from the "shifted" equation is much more accurate than the one obtained by applying the algorithm to the original equation. More precisely, it has been shown by Guo and Higham [8] that in order to achieve high accuracy it is necessary to use the singularity of M in the design of algorithms; otherwise, we can expect only to achieve an accuracy of $O(\sqrt{\varepsilon})$, where ε is the machine precision. With the use of the shift technique [9], the information on the singularity of M is plugged into the algorithm, and we may achieve full accuracy in the approximation as confirmed by the numerical experiments.

The paper is organized as follows. After some preliminaries presented in section 2, we show in section 3 how to reduce one step of Newton's iteration for (1.1) to the solution of a linear system with a structured matrix, and in section 4 we deal with the problem of solving such a system in $O(n^2)$ ops. In section 5 we show that the iteration proposed by Lu [14] shares the same displacement structure and thus its complexity can be reduced to $O(n^2)$ as well, and we exploit the connection between it and the Newton iteration. In section 6 we deal with the critical case where the Jacobian is singular, by using the shift technique. In section 7 we address the main numerical stability issues, and in section 8 we present some numerical examples. From the experiments performed so far, our method turns out to be much faster and accurate than the existing methods. Finally, in section 9, we discuss some possible generalizations of this algorithm together with some research lines.

2. Preliminaries. A basic tool which we use is the concept of a Cauchy-like matrix [4]. A matrix $C = (c_{ij})_{i,j=1,\dots,n}$ is called [illegible] if its elements are of the form $c_{ij} = \frac{u_i v_j}{r_i - s_j}$ for some constants u_i, v_i, r_i, s_i, $i = 1, \dots, n$, such that $r_i \neq s_j$ for each i, j. If we define $R = \operatorname{diag}(r_1, r_2, \dots, r_n)$ and $S = \operatorname{diag}(s_1, s_2, \dots, s_n)$, we have $RC - CS = uv^T$, where $u = [u_1, u_2, \dots, u_n]^T$ and $v = [v_1, v_2, \dots, v_n]^T$. The operator $C \mapsto RC - CS$ is called the [illegible], and u, v^T are called the [illegible] of C. Generalizing, if there exist two diagonal matrices R, S such that $RC - CS$ has rank r, we say that C has [illegible] r. When r is small with respect to the size of C, C is called a [illegible] with respect to the pair (R, S).

Note that, using (1.2), equation (1.1) can be rewritten as

$$XD + \Delta X = (Xq + e)(e^T + q^T X);$$

therefore any solution X is Cauchy-like with respect to $(\Delta, -D)$ and its generators are $u = Xq + e$ and $v^T = e^T + q^T X$.

We will also need some basic facts on M-matrices. A matrix $A = (a_{i,j}) \in \mathbb{R}^{n \times n}$ is called a Z [illegible] if $a_{ij} \leqslant 0$ for all $i \neq j$. A Z-matrix A is called an M [illegible] if there exists a nonnegative matrix B with spectral radius $\rho(B) = r$ such that $A = cI_n - B$ and $r \leqslant c$, where I_n is the identity matrix of order n.

The following results are well known and can be found in [1].

LEMMA 2.1. [illegible] Z [illegible] A [illegible]

1. A [illegible] M [illegible] $v > 0$ [illegible] $Av \geqslant 0$ [illegible] $w > 0$ [illegible] $w^T A \geqslant 0$
2. [illegible] A [illegible] A [illegible] M [illegible] $A^{-1} \geqslant 0$

LEMMA 2.2. [illegible] A [illegible] M [illegible] A [illegible] M [illegible]

Here and hereafter, inequalities on matrices and vectors are used in the componentwise sense.

Another useful tool is the Sherman–Morrison–Woodbury (SMW) matrix identity [5, p. 50].

LEMMA 2.3 (SMW formula). Let $D \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{k\times k}$, $U \in \mathbb{R}^{n\times k}$, $V \in \mathbb{R}^{k\times n}$. Then $D-UCV$ is nonsingular if and only if $C^{-1}-VD^{-1}U$ is nonsingular, and

$$(D - UCV)^{-1} = D^{-1} + D^{-1}U(C^{-1} - VD^{-1}U)^{-1}VD^{-1}.$$

The following lemma relates the SMW formula and M-matrices.

LEMMA 2.4. Let D, C, U, V satisfy the hypotheses of Lemma 2.3. Then D is a nonsingular M-matrix and C is a nonsingular M-matrix with $D, C, U, V \geqslant 0$ if and only if $D-UCV$ is (a nonsingular) M-matrix and $C^{-1} - VD^{-1}U$ is (a nonsingular) M-matrix.

Proof. Letting $C^{-1} - VD^{-1}U$ be a nonsingular M-matrix, the SMW formula yields

$$(D - UCV)^{-1} = D^{-1} + D^{-1}U(C^{-1} - VD^{-1}U)^{-1}VD^{-1},$$

and since all terms on the right-hand side are nonnegative, one has $(D - UCV)^{-1} > 0$, so $D-UCV$ is a nonsingular M-matrix by Lemma 2.1; the converse is analogous. By a continuity argument, the result can be extended to singular M-matrices. ∎

3. Newton's method. Newton's iteration applied to (1.1), for a suitable initial value $X^{(0)}$, generates the matrix sequence $\{X^{(k)}\}$ defined by the solution of the Sylvester equation [10]

$$(X^{(k+1)} - X^{(k)})(E - CX^{(k)}) + (A - X^{(k)}C)(X^{(k+1)} - X^{(k)}) = \mathcal{R}(X^{(k)}), \tag{3.1}$$

where $\mathcal{R}(X) = XCX - XE - AX + B$ is the Riccati operator. Using the Kronecker product notation, this can be written as

$$\operatorname{vec} X^{(k+1)} - \operatorname{vec} X^{(k)} = \left((E - CX^{(k)})^T \otimes I_n + I_n \otimes (A - X^{(k)}C)\right)^{-1} \operatorname{vec} \mathcal{R}(X^{(k)}), \tag{3.2}$$

where the vec operator stacks the columns of a matrix one above the other to form a single vector. Thus Newton's iteration is well defined when the matrix $M_{X^{(k)}} = (E - CX^{(k)})^T \otimes I_n + I_n \otimes (A - X^{(k)}C)$ is nonsingular for each k. With abuse of notation, we call the matrix M_X the Jacobian at X; in fact, it is the Jacobian matrix of the vector function $-\operatorname{vec} \circ R \circ \operatorname{vec}^{-1}$ at $\operatorname{vec}(X)$.

The following result, proved in [8] by Guo and Higham, provides sufficient conditions for the convergence of the Newton method.

THEOREM 3.1. Let

$$M = \begin{bmatrix} E & -C \\ -B & A \end{bmatrix}$$

be a nonsingular M-matrix and M be a singular irreducible M-matrix. Then $X^{(0)} = 0$ is such that (3.1) is a well-defined sequence $\{X^{(k)}\}$ converging to the minimal nonnegative solution of (1.1). Moreover, $M_{X^{(k)}} = (E - CX^{(k)})^T \otimes I + I \otimes (A - X^{(k)}C) \in \mathbb{R}^{n^2\times n^2}$ is a nonsingular M-matrix for $k \geqslant 0$.

Note that the problem stated in (1.2) satisfies the hypotheses of the previous theorem: in fact, we have

$$M = \begin{bmatrix} D & 0 \\ 0 & \Delta \end{bmatrix} - \begin{bmatrix} q \\ e \end{bmatrix} \begin{bmatrix} e^T & q^T \end{bmatrix},$$

and by Lemma 2.4 M is an M-matrix if and only if

$$0 \leqslant 1 - \begin{bmatrix} e^T & q^T \end{bmatrix} \begin{bmatrix} D^{-1} & 0 \\ 0 & \Delta^{-1} \end{bmatrix} \begin{bmatrix} q \\ e \end{bmatrix},$$

which reduces to

$$1 \geqslant e^T D^{-1} q + q^T \Delta^{-1} e = \sum_{i=1}^{n} \frac{c(1-\alpha)}{2} w_i + \sum_{i=1}^{n} \frac{c(1+\alpha)}{2} w_i = c, \tag{3.3}$$

in view of (1.3). This fact was also observed in [6].

In the following, we will consider a slightly more general case, i.e., when the matrix M is a generic diagonal plus rank-one matrix. Hence, (1.2) becomes

$$A = \Delta - \widetilde{e} q^T, \quad B = \widetilde{e} e^T, \quad C = \widetilde{q} q^T, \quad E = D - \widetilde{q} e^T, \tag{3.4}$$

where e, q, $\widetilde{e}$, $\widetilde{q}$ are any nonnegative vectors such that M, as defined in Theorem 3.1, is a nonsingular M-matrix or a singular irreducible M-matrix. Such generalization will prove useful when dealing with the critical case.

Observe that Newton's iteration for the coefficients of (1.1) defined in (3.4) can be rewritten as

$$\begin{aligned} X^{(k+1)} D + \Delta X^{(k+1)} = &-(X^{(k)} \widetilde{q} - X^{(k+1)} \widetilde{q})(q^T X^{(k)} - q^T X^{(k+1)}) \\ &+ (X^{(k+1)} \widetilde{q} + \widetilde{e})(e^T + q^T X^{(k+1)}); \end{aligned} \tag{3.5}$$

i.e., $X^{(k+1)}$ is a generalized Cauchy-like matrix with displacement rank 2. This property holds for all the iterates $X^{(k)}$, $k \geqslant 1$, of Newton's method obtained with any starting matrix $X^{(0)}$.

The Jacobian at $X^{(k)}$, in Kronecker product notation, takes the form

$$M_{X^{(k)}} = D^T \otimes I_n + I_n \otimes \Delta - (e + X^{(k)T} q) \widetilde{q}^T \otimes I_n - I_n \otimes (\widetilde{e} + X^{(k)} \widetilde{q}) q^T.$$

By setting $\mathcal{D} = D^T \otimes I_n + I_n \otimes \Delta$, $u^{(k)} = \widetilde{e} + X^{(k)} \widetilde{q}$, $v^{(k)} = e + X^{(k)T} q$, and

$$U^{(k)} = \begin{bmatrix} v^{(k)} \otimes I_n & I_n \otimes u^{(k)} \end{bmatrix}, \quad V = \begin{bmatrix} \widetilde{q}^T \otimes I_n \\ I_n \otimes q^T \end{bmatrix}, \tag{3.6}$$

we can rewrite $M_{X^{(k)}}$ as

$$M_{X^{(k)}} = \mathcal{D} - U^{(k)} V. \tag{3.7}$$

Since $U^{(k)} \in \mathbb{R}^{n^2 \times 2n}$ and $V \in \mathbb{R}^{2n \times n^2}$, the inversion of $M_{X^{(k)}}$ can be reduced to the inversion of a $2n \times 2n$ matrix using the SMW formula:

$$M_{X^{(k)}}^{-1} = \mathcal{D}^{-1} + \mathcal{D}^{-1} U^{(k)} (I_{2n} - V \mathcal{D}^{-1} U^{(k)})^{-1} V \mathcal{D}^{-1}. \tag{3.8}$$

This provides a new algorithm for implementing the Newton step, denoted by Algorithm 1, which relies on the function `fast_solve` for the fast solution of the system

$$\begin{aligned} &R^{(k)} x = b, \\ &R^{(k)} = I_{2n} - V \mathcal{D}^{-1} U^{(k)} \end{aligned} \tag{3.9}$$

function $X^{(k+1)}$=NewtonStep$(X^{(k)})$
$u^{(k)} = \widetilde{e} + X^{(k)} * \widetilde{q}$;
$v^{(k)} = e + X^{(k)T} * q$;
$\mathcal{R}(X^{(k)}) = u^{(k)} * v^{(k)T} - X^{(k)}D - \Delta X^{(k)}$;
$R_1 = [\widetilde{q}^T \otimes I_n\,; I_n \otimes q^T] * (\mathcal{D}^{-1} * \mathrm{vec}(\mathcal{R}(X^{(k)})))$;
$R_2 = \mathtt{fast\_solve}((I_{2n} - V\mathcal{D}^{-1}U^{(k)})R_2 = R_1)$;
$X^{(k+1)} = \mathcal{D}^{-1}(\mathrm{vec}(\mathcal{R}(X^{(k)})) + [v^{(k)} \otimes I_n \;\; I_n \otimes u^{(k)}] * R_2)$;
return $X^{(k+1)}$
end function

Algorithm 1: Fast Newton's step.

in $O(n^2)$ ops. The function `fast_solve` is described in the next section.

Note that since $\mathcal{D}$ is a diagonal matrix of size $n^2 \times n^2$, the matrix-vector product with matrix $\mathcal{D}^{-1}$ costs $O(n^2)$ ops, and the identities

$$(v^T \otimes I_n)\,\mathrm{vec}(W) = Wv,$$
$$(I_n \otimes v^T)\,\mathrm{vec}(W) = W^T v$$

allow one to compute the remaining products in $O(n^2)$ as well. Therefore the overall cost of Algorithm 1 is $O(n^2)$.

4. Fast Gaussian elimination for Cauchy-like matrices. We now address the problem of solving the linear system (3.9) given the vector b and the vectors $q, u^{(k)}, v^{(k)}$ such that

$$R^{(k)} = I_{2n} - \begin{bmatrix} \widetilde{q}^T \otimes I_n \\ I_n \otimes q^T \end{bmatrix} \mathcal{D}^{-1} \begin{bmatrix} v^{(k)} \otimes I_n & I_n \otimes u^{(k)} \end{bmatrix}. \tag{4.1}$$

First note that under the hypotheses of Theorem 3.1, $R^{(k)}$ is a nonsingular M-matrix by Lemma 2.4 applied to the nonsingular M-matrix $M_{X^{(k)}}$ of (3.7). Carrying out the products in (4.1) yields

$$R^{(k)} = I_{2n} - \begin{bmatrix} G^{(k)} & H^{(k)} \\ K^{(k)} & L^{(k)} \end{bmatrix} \tag{4.2}$$

with

$$\begin{aligned} G^{(k)} &= \mathrm{diag}(g_i^{(k)}), \quad & g_i^{(k)} &= \sum_{l=1}^{n} \frac{v_l^{(k)} \widetilde{q}_l}{d_l + \delta_i}, \\ H^{(k)} &= (h_{ij}^{(k)}), & h_{ij}^{(k)} &= \frac{u_i^{(k)} \widetilde{q}_j}{d_j + \delta_i}, \\ K^{(k)} &= (\kappa_{ij}^{(k)}), & \kappa_{ij}^{(k)} &= \frac{v_i^{(k)} q_j}{d_i + \delta_j}, \\ L^{(k)} &= \mathrm{diag}(l_i^{(k)}), \quad & l_i^{(k)} &= \sum_{l=1}^{n} \frac{u_l^{(k)} q_l}{d_i + \delta_l}. \end{aligned} \tag{4.3}$$

Thus $G^{(k)}$ and $L^{(k)}$ are diagonal, and $H^{(k)}$ and $K^{(k)}$ are Cauchy-like. Their displacement equations are

$$\Delta H^{(k)} + H^{(k)} D = u^{(k)} \widetilde{q}^T, \quad DK^{(k)} + K^{(k)} \Delta = v^{(k)} q^T.$$

Partition x and b according to the block structure of $R^{(k)}$ as $x = [x_1^T, x_2^T]^T$, $b = [b_1^T, b_2^T]^T$. Performing the block LU factorization of $R^{(k)}$ enables one to rewrite the system $R^{(k)} x = b$ as

$$\begin{bmatrix} I - G^{(k)} & -H^{(k)} \\ 0 & S^{(k)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ \widehat{b}_2 \end{bmatrix}, \tag{4.4}$$

where $S^{(k)} = I - L^{(k)} - K^{(k)}(I - G^{(k)})^{-1} H^{(k)}$ and $\widehat{b}_2 = b_2 - K^{(k)}(I - G^{(k)})^{-1} b_1$. The matrices $I - G^{(k)}$ and $S^{(k)}$ are nonsingular as they are a principal submatrix and the Schur complement of a nonsingular M-matrix, respectively. Moreover, $S^{(k)}$ enjoys the following displacement structure:

$$DS^{(k)} - S^{(k)} D = K^{(k)}(I - G^{(k)})^{-1} u^{(k)} \widetilde{q}^T - v^{(k)} q^T (I - G^{(k)})^{-1} H^{(k)}.$$

This can be easily proved since D, Δ, $I - G^{(k)}$, $I - L^{(k)}$ all commute because they are diagonal; in fact,

$$\begin{aligned} DS^{(k)} &= D(I - L^{(k)}) - DK^{(k)}(I - G^{(k)})^{-1} H^{(k)} \\ &= (I - L^{(k)})D + (K^{(k)}\Delta - v^{(k)} q^T)(I - G^{(k)})^{-1} H^{(k)} \\ &= (I - L^{(k)})D + K^{(k)}(I - G^{(k)})^{-1} \Delta H^{(k)} - v^{(k)} q^T (I - G^{(k)})^{-1} H^{(k)} \\ &= (I - L^{(k)})D - K^{(k)}(I - G^{(k)})^{-1}(H^{(k)} D - u^{(k)} \widetilde{q}^T) - v^{(k)} q^T (I - G^{(k)})^{-1} H^{(k)} \\ &= S^{(k)} D + K^{(k)}(I - G^{(k)})^{-1} u^{(k)} \widetilde{q}^T - v^{(k)} q^T (I - G^{(k)})^{-1} H^{(k)}. \end{aligned}$$

Thus $S^{(k)}$ is a generalized Cauchy-like matrix with displacement rank 2 with respect to the singular operator $DS^{(k)} - S^{(k)} D$. We can use this property to develop an ad hoc variation of the Gohberg–Kailath–Olshevsky (GKO) algorithm for the fast LU factorization of matrices with displacement structure [4]. The GKO algorithm, for a generalized Cauchy-like matrix S with generators M_1 and N_1, essentially goes on as follows (ignoring pivoting for the sake of simplicity):

1. From the generators M_1, N_1 of

   $$S = \begin{bmatrix} d_1 & u_1 \\ l_1 & S_2 \end{bmatrix},$$

   such that $DS - SD = M_1 N_1$, recover the first row and the first column of S and store them as the first column of L,

   $$\begin{bmatrix} 1 \\ \frac{l_1}{d_1} \end{bmatrix},$$

   and the first row of U,

   $$\begin{bmatrix} d_1 & u_1 \end{bmatrix},$$

   in the LU factorization of S.

2. Compute the generators M_2, N_2 of the Schur complement $S_2 - \frac{l_1 u_1}{d_1}$ as

$$M_2 = M_{12} - \frac{l_1}{d_1} m_{11}, \quad N_2 = N_{12} - n_{11} \frac{u_1}{d_1},$$

where

$$M_1 = \begin{bmatrix} m_{11} \\ M_{12} \end{bmatrix}, \quad N_1 = \begin{bmatrix} n_{11} & N_{12} \end{bmatrix}.$$

3. Apply the algorithm recursively to compute the LU factorization $L_2 U_2$ of the Schur complement $S_2 - \frac{l_1 u_1}{d_1}$; then reconstruct the factors

$$L = \begin{bmatrix} 1 & \frac{l_1}{d_1} \\ 0 & L_2 \end{bmatrix}, \quad U = \begin{bmatrix} d_1 & u_1 \\ 0 & U_2 \end{bmatrix}.$$

The problem in our context is that d_1 cannot be retrieved from the generators, due to the singularity of the operator $S \mapsto DS - SD$. In fact, it is easy to see that the null space of $DS - SD$ is the set of all diagonal matrices. Thus, we need a different method to compute and update the diagonal elements of S through the LU factorization. Our approach consists in storing the main diagonal of S in a vector s and updating it at each step of the Gaussian elimination as if we were performing a customary (nonstructured) Gaussian elimination. This can be achieved at the general step k by using the relation $S_{ii} \leftarrow S_{ii} - L_{ik} U_{ki}$. Since we have to update only n elements at each step, the overhead of updating the diagonal is $O(n^2)$, and thus the complete algorithm requires $O(n^2)$ ops. A simple implementation, which includes partial pivoting, is given in Algorithm 2 and requires $10n^2$ ops.

```
function [PL,U]=fastPLU(d,s,M,N)
u=[1,1,...,1]'; L=U=zeros(n,n);
for k=1:n
 L_ik=(∑_j M_ij N_jk)/(d_i−d_k) for all i≠k such that u_i=1;
 L_kk=s_k if u_i=1;
 choose p such that |L_pk|=max_i |L_ik|;
 u_p=0;
 U_kk=L_pk;
 L_ik=L_ik/U_kk for all i such that u_i=1
 U_kj=(∑_i M_pi N_ij)/(d_p−d_j) for all j=k+1,...,n, j≠p;
 U_kp=s_p;
 M_ij=M_ij − L_ik M_pj for all j, i such that u_i=1;
 N_ij=N_ij − N_ik U_kj/U_kk for all i=1,...,n, j=k+1,...,n;
 s_i=s_i − L_ik U_ki for all i such that u_i=1;
 return L,U;
end function
```

Algorithm 2: Fast LU factorization.

Using this algorithm, complemented with back-substitution, provides an implementation of the function `fast_solve`$(R^{(k)}x = b)$ of complexity $O(n^2)$ that was used in Algorithm 1.

5. Lu's iteration. Lu [14] proposed a different approach for solving the Riccati equation (1.1) when the coefficients are in the form (3.4). The idea is applying Newton's iteration to an equation involving the displacement generators $u = X\widetilde{q} + \widetilde{e}$ and $v = X^T q + e$ of the solution X. His algorithm can be expressed as the following iteration for the sequences $\{\widehat{u}^{(k)}\}$, $\{\widehat{v}^{(k)}\}$, $k \geqslant -1$:

$$\begin{bmatrix} \widehat{u}^{(k+1)} \\ \widehat{v}^{(k+1)} \end{bmatrix} = (\widehat{R}^{(k)})^{-1} \begin{bmatrix} \widetilde{e} - \widehat{H}^{(k)}\widehat{v}^{(k)} \\ e - \widehat{K}^{(k)}\widehat{u}^{(k)} \end{bmatrix}, \tag{5.1}$$

starting from $\widehat{u}^{(-1)} = \widehat{v}^{(-1)} = 0$ (as we will see later on, indexing from $k = -1$ will simplify the subsequent analysis). Here $\widehat{R}^{(k)}$, $\widehat{H}^{(k)}$, and $\widehat{K}^{(k)}$ are defined as

$$\widehat{R}^{(k)} = I_{2n} - \begin{bmatrix} \widehat{G}^{(k)} & \widehat{H}^{(k)} \\ \widehat{K}^{(k)} & \widehat{L}^{(k)} \end{bmatrix}, \tag{5.2}$$

$$\begin{aligned} \widehat{G}^{(k)} &= \operatorname{diag}(\widehat{g}_i^{(k)}), & \widehat{g}_i^{(k)} &= \sum_{l=1}^{n} \frac{\widehat{v}_l^{(k)}\widetilde{q}_l}{d_l + \delta_i}, \\ \widehat{H}^{(k)} &= (\widehat{h}_{ij}^{(k)}), & \widehat{h}_{ij}^{(k)} &= \frac{\widehat{u}_i^{(k)}\widetilde{q}_j}{d_j + \delta_i}, \\ \widehat{K}^{(k)} &= (\widehat{\kappa}_{ij}^{(k)}), & \widehat{\kappa}_{ij}^{(k)} &= \frac{\widehat{v}_i^{(k)} q_j}{d_i + \delta_j}, \\ \widehat{L}^{(k)} &= \operatorname{diag}(\widehat{l}_i^{(k)}), & \widehat{l}_i^{(k)} &= \sum_{l=1}^{n} \frac{\widehat{u}_l^{(k)} q_l}{d_i + \delta_l}, \end{aligned} \tag{5.3}$$

which are, formally, the same relations as in (4.2) and (4.3).

As a first result, since both algorithms are based on the solution of a system with the same structure, we obtain that Algorithm 2 can also be used in the implementation of Lu's iteration to reduce its computational cost to $O(n^2)$. But there is a deeper connection between the two algorithms.

THEOREM 5.1. [illegible] $\{\widehat{u}^{(k)}\}$ $\{\widehat{v}^{(k)}\}$ $k \geqslant -1$ [illegible] (1.1) [illegible] (3.4) [illegible] $\{X^{(k)}\}$ $k \geqslant 0$ [illegible] $X^{(0)} = 0$ [illegible] $k \geqslant 0$ [illegible]

$$\widehat{u}^{(k)} = X^{(k)}\widetilde{q} + \widetilde{e}, \quad \widehat{v}^{(k)} = X^{(k)T} q + e.$$

[illegible]. We will prove the result by induction over k. It is easy to check from the definitions that $\widehat{R}^{(-1)} = I_{2n}$, and thus $\widehat{u}^{(0)} = \widetilde{e}$, $\widehat{v}^{(0)} = e$; therefore the base step $k = 0$ holds. As a side note, this means that we can save an iteration by starting the computation from $u^{(0)} = \widetilde{e}$, $v^{(0)} = e$.

Assuming by induction that $\widehat{u}^{(k)} = X^{(k)}\widetilde{q} + \widetilde{e} = u^{(k)}$, $\widehat{v}^{(k)} = X^{(k)T} q + e = v^{(k)}$, we find that (4.3) and (5.2) define the same matrices; therefore, from now on, we will drop the superscript (k) and the hat symbol to ease the notation.

We have

$$V\mathcal{D}^{-1} \operatorname{vec}\mathcal{R}(X) = V\mathcal{D}^{-1} \operatorname{vec}(uv^T) - V \operatorname{vec} X = \begin{bmatrix} \widetilde{e} - (I - G)u \\ e - (I - L)v \end{bmatrix},$$

in view of the relations

$$\mathcal{D}^{-1}\operatorname{vec}(XD+\Delta X)=\operatorname{vec}X,$$
$$V\mathcal{D}^{-1}\operatorname{vec}(uv^T)=\begin{bmatrix} Gu \\ Lv \end{bmatrix},$$

which can be easily verified from the definitions of $\mathcal{D}$, G, and L, where V is the matrix defined in (3.6).

Applying the operator V to both sides of (3.2) yields

(5.4)
$$\begin{aligned} V\operatorname{vec}(X^{(k+1)}-X) &= V\mathcal{D}^{-1}\operatorname{vec}\mathcal{R}(X)+V\mathcal{D}^{-1}U(I_{2n}-V\mathcal{D}^{-1}U)^{-1}V\mathcal{D}^{-1}\operatorname{vec}\mathcal{R}(X)\\ &= (I_{2n}+V\mathcal{D}^{-1}U(I_{2n}-V\mathcal{D}^{-1}U)^{-1})V\mathcal{D}^{-1}\operatorname{vec}\mathcal{R}(X)\\ &= (I_{2n}-V\mathcal{D}^{-1}U)^{-1}V\mathcal{D}^{-1}\operatorname{vec}\mathcal{R}(X), \end{aligned}$$

where the last equation holds since $I+M(I-M)^{-1}=(I-M)^{-1}$.

We recall that $R=(I_{2n}-V\mathcal{D}^{-1}U)$ and

$$\begin{bmatrix} \widehat{u}^{(k+1)} \\ \widehat{v}^{(k+1)} \end{bmatrix} = R^{-1}\begin{bmatrix} \widetilde{e}-Hu \\ e-Kv \end{bmatrix}$$

(the latter being Lu's iteration). Now we can explicitly compute

$$\begin{aligned} R\begin{bmatrix} \widehat{u}^{k+1}-u \\ \widehat{v}^{k+1}-v \end{bmatrix} &= \begin{bmatrix} \widetilde{e}-Hu \\ e-Kv \end{bmatrix}-\left(I_{2n}-\begin{bmatrix} G & H \\ K & L \end{bmatrix}\right)\begin{bmatrix} u \\ v \end{bmatrix}\\ &= \begin{bmatrix} \widetilde{e}-Hv \\ e-Ku \end{bmatrix}-\begin{bmatrix} u \\ v \end{bmatrix}+\begin{bmatrix} Gu+Hv \\ Ku+Lv \end{bmatrix}=\begin{bmatrix} \widetilde{e}-(I-G)u \\ e-(I-L)v \end{bmatrix}=V\mathcal{D}^{-1}\operatorname{vec}\mathcal{R}(X), \end{aligned}$$

and substitute it into (5.4) to get

$$V\operatorname{vec}(X^{(k+1)}-X)=\begin{bmatrix} \widehat{u}^{k+1}-u \\ \widehat{v}^{k+1}-v \end{bmatrix}.$$

Finally, using the definition of V in (3.6), we find that

$$\begin{bmatrix} \widehat{u}^{k+1}-u \\ \widehat{v}^{k+1}-v \end{bmatrix}=V\operatorname{vec}(X^{(k+1)}-X)=\begin{bmatrix} X^{(k+1)}\widetilde{q}-X\widetilde{q} \\ X^{(k+1)T}q-X^Tq \end{bmatrix}=\begin{bmatrix} X^{(k+1)}\widetilde{q}+\widetilde{e}-u \\ X^{(k+1)T}q+e-v \end{bmatrix}$$

and thus $\widehat{u}^{k+1}=X^{(k+1)}\widetilde{q}+\widetilde{e}$, $\widehat{v}^{k+1}=X^{(k+1)T}q+e$. □

The theorem brings deeper insight into Newton's and Lu's iterations. For example, Theorem 6 of [14], which states that Lu's iteration is well defined and converges monotonically to the minimal solution of the NARE, can now be seen as a special case of Theorem 3.1. Moreover, Lu's iteration can be viewed as a structured Newton's iteration exploiting the displacement structure found in (3.5). Therefore, the two algorithms take the same number of iterations to converge, as the computation they perform is the same. Observe also that Lu's version of this algorithm is slightly faster, since it updates only the $2n$ entries of the generators of the matrix $\{X^{(k)}\}$ instead of all the n^2 entries. For this reason, we will present only numerical results regarding Lu's iteration.

6. Shift technique. In the case where $(c, \alpha) = (1, 0)$, the Jacobian M_X appearing in the Newton iteration is singular when X is the solution of the NARE. We refer to this as the [illegible]. Several drawbacks are encountered in the critical case; see the analysis of Guo and Higham in [8] for more details. The singularity of the Jacobian does not guarantee the quadratic convergence of Newton's iteration; in fact, Newton's and therefore Lu's method converge linearly. Moreover, a perturbation $O(\varepsilon)$ in the coefficients of the equation leads to an $O(\sqrt{\varepsilon})$ variation in the solution.

These drawbacks can be easily removed by means of the shift technique originally introduced by He, Meini, and Rhee in [12] and applied to Riccati equations in [9] and [2].

A characterization of the critical case can be given in terms of the eigenvalues of the matrix

$$H = \begin{bmatrix} E & -C \\ B & -A \end{bmatrix}, \tag{6.1}$$

obtained by premultiplying the M-matrix M defined in (1.4) by the matrix $J = \operatorname{diag}(I_n, -I_n)$. In fact, the matrix H has a double zero eigenvalue corresponding to a 2×2 Jordan block (see [9] and the references therein).

The shift technique, as described in [9], consists in a rank-one correction to the matrix H of (6.1) which gives $\widetilde{H} = H + \eta v p^T$, where $\eta > 0$, v is a right eigenvector of H corresponding to the zero eigenvalue, and p is an arbitrary vector such that $p^T v = 1$.

The nice feature of this transformation is that the Riccati equation associated with the matrix $\widetilde{H}$ has the same minimal solution as the original one, although the new Jacobian matrix at the solution is not singular. This removes the above-mentioned drawbacks. Now the point is to show that it is still possible to provide a fast implementation of Newton's iteration for the new equation obtained by means of the shift technique. This is the goal of this section.

Under the assumptions (1.2), (1.3) a right eigenvector of H corresponding to zero is $v = \begin{bmatrix} v_1^T & v_2^T \end{bmatrix}^T$, where $v_1 = D^{-1}q$, $v_2 = \Delta^{-1}e$. This can be seen by direct inspection using the fact that $e^T D^{-1} q + q^T \Delta^{-1} e = c = 1$ (see (3.3)).

The rank-one correction we construct is

$$\widetilde{H} = H + \eta \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} p^T,$$

where $0 < \eta \leqslant d_1$ and $p^T = \begin{bmatrix} e^T & q^T \end{bmatrix}$. It holds that $p^T v = 1$; in fact, $p^T v = e^T D^{-1} q + q^T \Delta^{-1} e = 1$. It is proved in [9] that $\widetilde{H}$ has a simple zero eigenvalue.

The matrix $\widetilde{H}$ defines the new Riccati equation

$$X\widetilde{C}X - X\widetilde{E} - \widetilde{A}X + \widetilde{B} = 0, \tag{6.2}$$

with

$$\widetilde{A} = A - \eta v_2 q^T, \quad \widetilde{B} = B + \eta v_2 e^T, \quad \widetilde{C} = C - \eta v_1 q^T, \quad \widetilde{E} = E + \eta v_1 e^T. \tag{6.3}$$

It is proved in [9] that the minimal nonnegative solution of (1.1) is the minimal nonnegative solution of (6.2).

With the choice of $p^T = [e^T \; q^T]$, $\widetilde{H}$ remains a diagonal plus rank-one matrix as well as $\widetilde{M} = J\widetilde{H}$; hence, we need only prove that $\widetilde{M}$ is an M-matrix to ensure that

the algorithms proposed in sections 3 and 5 can be applied to (6.2). In fact, we have

$$\widetilde{M} = \begin{bmatrix} D & 0 \\ 0 & \Delta \end{bmatrix} - \begin{bmatrix} q - \eta v_1 \\ e + \eta v_2 \end{bmatrix} \begin{bmatrix} e^T & q^T \end{bmatrix},$$

and since we chose $0 < \eta \leqslant d_1 < d_2 < \cdots < d_n$, and $q \geqslant 0$, it holds that $q - \eta v_1 = (I_n - \eta D^{-1})q \geqslant 0$; thus $\widetilde{M}$ is a Z-matrix. By the Perron–Frobenius theorem applied to $\rho I - M$, there exists a vector $u > 0$ such that $u^T M = 0$, and in the critical case we have $u^T J v = 0$ (observe that $u^T J$ is a left eigenvector of $H = JM$ corresponding to the zero eigenvalue and recall that right and left eigenvectors corresponding to the same eigenvalue in a Jordan block of dimension $n \geqslant 2$ are orthogonal); therefore,

$$u^T \widetilde{M} = u^T M + \eta u^T J v p^T = 0,$$

and thus by part 1 of Lemma 2.1 $\widetilde{M}$ is an M-matrix.

In this way, Newton's iteration applied to (6.2) provides a quadratically convergent algorithm of complexity $O(n^2)$ for solving the Riccati equation (1.1) in the critical case. Moreover, since the singularity has been removed, it is expected that X, as minimal solution of (6.2), is better conditioned with respect to the coefficients of (1.1), and that a higher precision can be reached in the computed solution. This fact is confirmed by the numerical experiments as shown in section 8.

7. Numerical stability. Our first concern about numerical stability is proving that the matrix $R^{(k)} = I_{2n} - V\mathcal{D}^{-1}U^{(k)}$ resulting after the application of the SMW formula to the Jacobian $\mathcal{D} - U^{(k)}V$ is well-conditioned whenever the Jacobian is. In the following analysis, we will assume that the norm $\left\|(\mathcal{D} - U^{(k)}V)^{-1}\right\|_1$ is bounded, and we will drop the superscripts (k) to simplify the notation.

Observe that $0 \leqslant \mathcal{D}^{-1} \leqslant (\mathcal{D} - UV)^{-1}$; therefore $\mathcal{D}$ is well-conditioned. Moreover, one has

$$\mathcal{B}^{-1} = \begin{bmatrix} (\mathcal{D} - UV)^{-1} & 0 \\ V(\mathcal{D} - UV)^{-1} & I \end{bmatrix} \begin{bmatrix} I & U \\ 0 & I \end{bmatrix}, \quad \text{with} \quad \mathcal{B} = \begin{bmatrix} \mathcal{D} & -U \\ -V & I \end{bmatrix};$$

therefore $\mathcal{B}$ is an M-matrix and is well-conditioned. Now, $R = I - V\mathcal{D}^{-1}U$ is the Schur complement of $\mathcal{D}$ in $\mathcal{B}$, and thus R^{-1} is a submatrix of $\mathcal{B}^{-1}$ [5]. This implies $\left\|R^{-1}\right\|_1 \leqslant \left\|\mathcal{B}^{-1}\right\|_1$; hence R is well-conditioned, too.

Another stability problem could arise from the generators' growth during the fast Gaussian elimination step. Generator growth has been reported in some cases with the GKO algorithm [17], especially when the starting generators are ill-conditioned. This is not our case, since the starting generators are bounded, and no significant generator growth has been observed during our experiments.

8. Numerical experiments. We consider the numerical examples suggested in [10] and used also in [14]. The sequences t_i and w_i, which appear in the discretization as the nodes and weights of a Gaussian quadrature method, are obtained by dividing the interval $[0, 1]$ into $n/4$ subintervals of equal length and by applying to each one the 4-node Gauss–Legendre quadrature.

The computation has been performed with three different choices of the parameters (c, α), namely, $(0.5, 0.5)$, $(1 - 10^{-6}, 10^{-8})$, and $(1, 0)$. The latter is the critical case, and thus the quadratic convergence of Newton's method is not guaranteed. In this case, the algorithms are more prone to numerical problems, since the matrices to be inverted are near-to-singular.

The algorithms have been implemented in Fortran 90, and the tests have been carried out using the Lahey Fortran compiler on a Xeon biprocessor with 2.8 GHz. We have compared Lu's algorithm presented in [14] with its fast version based on Algorithm 2. In the critical case, we have also made a comparison with the shifted algorithm of section 6. To compute the step (5.1) of Lu's algorithm we have solved a linear system using the LAPACK `la_gesv` function.

TABLE 8.1

Comparison of CPU time in seconds of Lu's algorithm (Lu), its fast version presented here (LuF), and the shifted algorithm in the critical case (LuFS).

	$\alpha = 0.5$, $c = 0.5$		$\alpha = 10^{-8}$, $c = 1 - 10^{-6}$		$\alpha = 0$, $c = 1$		
n	Lu	LuF	Lu	LuF	Lu	LuF	LuFS
32	0.002	0.001	0.005	0.002	0.007	0.004	0.001
64	0.009	0.002	0.028	0.009	0.040	0.015	0.005
128	0.050	0.010	0.175	0.034	0.290	0.054	0.015
256	0.401	0.053	1.369	0.167	2.253	0.278	0.074
512	4.125	0.343	14.63	1.109	21.065	1.948	0.507
1024	40.5	1.456	141.3	4.959	212.6	7.957	2.251
2048	327	5.785	1146	19.914	1850	32.478	9.061
4096	2775	28.503	9669	89.78	15974	147.3	40.917

TABLE 8.2

Comparison of the relative error (and in parentheses the number of steps) of Lu's algorithm (Lu), its fast version presented here (LuF), and the shifted algorithm in the critical case (LuFS).

	$\alpha = 0.5$, $c = 0.5$	
n	Lu	LuF
32	$4.8 \cdot 10^{-16}$ (4)	$2.3 \cdot 10^{-16}$ (5)
256	$1.6 \cdot 10^{-15}$ (4)	$4.0 \cdot 10^{-16}$ (5)

	$\alpha = 0$, $c = 1$		
n	Lu	LuF	LuFS
32	$5.2 \cdot 10^{-8}$ (25)	$4.2 \cdot 10^{-8}$ (26)	$4.4 \cdot 10^{-16}$ (6)
256	$4.6 \cdot 10^{-8}$ (25)	$8.0 \cdot 10^{-8}$ (25)	$1.2 \cdot 10^{-15}$ (6)

In Table 8.1 we compare the timing of Lu's algorithm, which has a computational cost of $O(n^3)$ ops, with that of its fast version, which costs $O(n^2)$. The numerical results highlight the different order of complexity. Observe that in the critical case the shift technique reduces the timings even further.

In Table 8.2 we compare the relative error of the two methods and the number of steps required. Here the error is computed as $\|\widetilde{X} - X\|_1 / \|X\|_1$, where $\widetilde{X}$ and X are the solution computed in double and in quadruple precision, respectively.

The stopping criterion is based on the computation of

$$\mathrm{Res} = \frac{\|u_k - u_{k-1}\|_1 + \|v_k - v_{k-1}\|_1}{2}.$$

Observe that the cost of computing Res is negligible.

As one can see, in the critical case the accuracy of the solution obtained with the nonshifted algorithms is of the order of $O(\sqrt{\varepsilon})$, where ε is the machine precision, in strict accordance with [8]. The speedup obtained is greater than 2 even for small values of n. In the critical case with size $n = 4096$ our algorithm is about 390 times faster than Lu's original algorithm. The problems deriving from the large number of steps and the poor accuracy are completely removed by the shift technique.

9. Generalizations and future work. Our algorithm can be easily extended to the case where M is diagonal plus rank k.

A challenging issue is to prove that Lu's iteration for this problem can be effectively computed with less than $O(n^2)$ ops. Actually, the literature provides algorithms for computing the Cauchy matrix-vector product [3] and for approximating the inverse of a Cauchy matrix [16] with $O(n \log^2 n)$ ops (, , ,).

However, the bottleneck for this improvement is computing the product of $\mathcal{D}^{-1}$ by the vector $v = \mathrm{vec}(\mathcal{R}(X^{(k)}))$, which apparently requires n^2 multiplications. In fact, the size of $\mathcal{D}$ is n^2. Observe that this problem concerns only the algorithm presented in section 3. In fact, Lu's version of Newton's iteration shown in section 5 is expressed through matrix-vector products and solutions of linear systems where the involved matrices are Cauchy-like.

If the superfast algorithms can be adapted to deal with singular displacement operators and if the approximation and the numerical errors introduced do not destroy the quadratic convergence of Newton's method, then the computational cost of Lu's iteration could be reduced further.

Another interesting issue is the acceleration of existing algorithms like the (shifted) structure-preserving doubling algorithm [9, 11] or the (shifted) logarithmic and cyclic reduction [2, 7], relying once again on the specific structure of the problem.

Acknowledgments. The authors are indebted to Luca Gemignani and Beatrice Meini for the many stimulating discussions which helped to improve the presentation of the paper. The authors wish to thank the anonymous referees for their useful comments and remarks.

REFERENCES

[1] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.

[2] D. A. Bini, B. Iannazzo, G. Latouche, and B. Meini, *On the solution of algebraic Riccati equations arising in fluid queues*, Linear Algebra Appl., 413 (2006), pp. 474–494.

[3] A. Gerasoulis, *A fast algorithm for the multiplication of generalized Hilbert matrices with vectors*, Math. Comp., 50 (1988), pp. 179–188.

[4] I. Gohberg, T. Kailath, and V. Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp., 64 (1995), pp. 1557–1576.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins Studies in the Mathematical Sciences, The Johns Hopkins University Press, Baltimore, MD, 1996.

[6] C.-H. Guo, *Nonsymmetric algebraic Riccati equations and Wiener–Hopf factorization for M-matrices*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 225–242.

[7] C.-H. Guo, *Efficient methods for solving a nonsymmetric algebraic Riccati equation arising in stochastic fluid models*, J. Comput. Appl. Math., 192 (2006), pp. 353–373.

[8] C.-H. Guo and N. J. Higham, *Iterative solution of a nonsymmetric algebraic Riccati equation*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 396–412.

[9] C.-H. Guo, B. Iannazzo, and B. Meini, *On the doubling algorithm for a (shifted) nonsymmetric algebraic Riccati equation*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1083–1100.

[10] C.-H. Guo and A. J. Laub, *On the iterative solution of a class of nonsymmetric algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 376–391.

[11] X.-X. Guo, W.-W. Lin, and S.-F. Xu, *A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equation*, Numer. Math., 103 (2006), pp. 393–412.

[12] C. He, B. Meini, and N. H. Rhee, *A shifted cyclic reduction algorithm for quasi-birth-death problems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 673–691.

[13] J. Juang and W.-W. Lin, *Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 228–243.

[14] L.-Z. Lu, *Newton iterations for a non-symmetric algebraic Riccati equation*, Numer. Linear Algebra Appl., 12 (2005), pp. 191–200.

[15] L.-Z. Lu, *Solution form and simple iteration of a nonsymmetric algebraic Riccati equation arising in transport theory*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 679–685.

[16] P.-G. Martinsson, V. Rokhlin, and M. Tygert, *A fast algorithm for the inversion of general Toeplitz matrices*, Comput. Math. Appl., 50 (2005), pp. 741–752.

[17] D. R. Sweet and R. P. Brent, *Error analysis of a fast partial-pivoting method for structured matrices*, in Advanced Signal Processing Algorithms, Proceedings of SPIE 2563, F. T. Luk, ed., SPIE, Bellingham, WA, 1995, pp. 266–280.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 291–311

ON ASYMPTOTIC CONVERGENCE OF NONSYMMETRIC JACOBI ALGORITHMS*

CHRISTIAN MEHL†

Abstract. The asymptotic convergence behavior of cyclic versions of the nonsymmetric Jacobi algorithm for the computation of the Schur form of a general complex matrix is investigated. Similar to the symmetric case, the nonsymmetric Jacobi algorithm proceeds by applying a sequence of rotations that annihilate a pivot element in the strict lower triangular part of the matrix until convergence to the Schur form of the matrix is achieved. In this paper, it is shown that the cyclic nonsymmetric Jacobi method converges locally and asymptotically quadratically under mild hypotheses if special ordering schemes are chosen, namely, ordering schemes that lead to so-called northeast directed sweeps. The theory is illustrated by the help of numerical experiments. In particular, it is shown that there are ordering schemes that lead to asymptotic quadratic convergence for the cyclic symmetric Jacobi method, but only to asymptotic linear convergence for the cyclic nonsymmetric Jacobi method. Finally, a generalization of the nonsymmetric Jacobi method to the computation of the Hamiltonian Schur form for Hamiltonian matrices is introduced and investigated.

Key words. Schur form, nonsymmetric Jacobi algorithm, asymptotic convergence, Hamiltonian Jacobi algorithm, Hamiltonian Schur form

AMS subject classification. 65F15

DOI. 10.1137/060663246

1. Introduction.

Jacobi's method [21] for the diagonalization of a symmetric matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ is a famous, successful, and easy-to-implement algorithm for the computation of eigenvalues of symmetric matrices. No wonder that this algorithm has been generalized or adapted to many other classes of matrices; see, among other references, [1, 3, 5, 9, 11, 12, 13, 15, 16, 29, 34]. In this paper, we will focus on a particular cyclic Jacobi-like algorithm for the computation of the Schur form of a complex matrix. The basic idea of the algorithm is a direct adaption of Jacobi's method to the nonsymmetric case that was proposed in 1955 by Greenstadt [13] and later was taken up and modified by various authors [4, 10, 14, 19, 24, 32]. Given a matrix $M = (m_{ij}) \in \mathbb{C}^{n\times n}$, the algorithm selects in each step a pivot element m_{kl}, $k < l$ in the strict lower triangular part. Then a similarity transformation with a rotation $U = (u_{ij}) \in \mathbb{C}^{n\times n}$ is applied to M that annihilates the entry m_{kl} of M:

$$\begin{bmatrix} 1 & & & & & \\ & \overline{u}_{kk} & & \overline{u}_{lk} & & \\ & & 1 & & & \\ & & & 1 & & \\ & \overline{u}_{kl} & & \overline{u}_{ll} & & \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} * & * & * & * & * & * \\ \cdot & * & * & * & * & * \\ \cdot & \cdot & * & * & * & * \\ \cdot & \cdot & \cdot & * & * & * \\ \cdot & m_{kl} & \cdot & \cdot & * & * \\ \cdot & \cdot & \cdot & \cdot & \cdot & * \end{bmatrix} \begin{bmatrix} 1 & & & & & \\ & u_{kk} & & u_{kl} & & \\ & & 1 & & & \\ & & & 1 & & \\ & u_{lk} & & u_{ll} & & \\ & & & & & 1 \end{bmatrix} = \begin{bmatrix} * & * & * & * & * & * \\ \cdot & * & * & * & * & * \\ \cdot & \cdot & * & * & * & * \\ \cdot & \cdot & \cdot & * & * & * \\ \cdot & 0 & \cdot & \cdot & * & * \\ \cdot & \cdot & \cdot & \cdot & \cdot & * \end{bmatrix}$$

Here, U coincides with the identity except for the elements $u_{kk} = u_{ll} = \cos x$, and $u_{kl} = -e^{-i\alpha} \sin x$, $u_{lk} = e^{i\alpha} \sin x$ for some $x, \alpha \in \mathbb{R}$.

*Received by the editors June 20, 2006; accepted for publication (in revised form) by V. Simoncini August 28, 2007; published electronically March 19, 2008. This research was partially supported by Deutsche Forschungsgemeinschaft through Matheon, the DFG Research Center "Mathematics for key technologies" in Berlin.

http://www.siam.org/journals/simax/30-1/66324.html

†School of Mathematics, Watson Building, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK (mehl@maths.bham.ac.uk).

Generically, there are two choices for the transformation matrix U, and throughout this paper, we will always choose the transformation matrix that is closest to the identity. This corresponds to choosing always the rotation matrix with the smaller rotation angle, a fact that is crucial for the proof of asymptotic quadratic convergence of Jacobi's algorithm for symmetric matrices. Concerning global convergence, this strategy need not be the best. Actually, convergence can be accelerated by sorting the diagonal entries of the occurring 2×2 submatrices, a technique that sometimes requires the application of the rotation corresponding to the larger angle. In this paper, however, we focus on the asymptotic convergence behavior and, therefore, we restrict ourselves to the use of rotations corresponding to the smaller angle.

Cyclic version of Jacobi's algorithm use a fixed sequence of pivot elements to be annihilated in that order. If every element in the strict lower triangular part is annihilated at least once; then the sequence of corresponding Jacobi steps on the matrix is called a [illegible], and it is repeated until convergence has been achieved.

In contrast to Jacobi's original algorithm, its generalization to general complex matrices has not yet become a well-known and widely used algorithm, probably mainly for two reasons. First, neither global nor local convergence proofs of the method could be given so far, although convergence has been observed in numerical experiments. (In [19], Huang proved convergence of the method for the case $n = 3$, but the case $n > 3$ remains an open problem.) Second, the algorithm converges slowly and is expensive. A flop count reveals that three sweeps of the cyclic nonsymmetric Jacobi method are approximately as expensive as a full run of the QR algorithm as implemented in Matlab. Moreover, the average number of sweeps needed for convergence for a 70×70 matrix is already as high as 30, making the method 10 times more expensive then the QR algorithm in that case. Since the average number of sweeps needed for convergence even increases with increasing dimension of the matrix, the algorithm seemed to be out of competition.

However, in recent years there has been renewed interest in nonsymmetric Jacobi methods, because they can easily be adapted to the solution of structured eigenvalue problems, e.g., of Hamiltonian eigenvalue problems [1], for generalized Hermitian eigenvalue problems [27], for palindromic eigenvalue problems [18, 25], and for doubly structured eigenvalue problems [11]. For those eigenvalue problems, a satisfactory generalization of the QR algorithm is not available due to the lack of a corresponding reduction to a structured version of the Hessenberg form in finitely many steps. (This phenomenon is known as "Van Loan's curse" [6].) Thus, the competition for other algorithms is open again. In fact, a variant of the nonsymmetric Jacobi method designed for the solution of the generalized palindromic eigenvalue problem (i.e., the eigenvalue problem $\lambda Ax = A^T x$) has been successfully used in [25] as an ingredient for a structure-preserving eigensolver, where it was applied to the solution of small (i.e., of size $\mathcal{O}(10)$) eigenvalue problems with eigenvalues close to the unit circle.

On the other hand, the nonsymmetric Jacobi method converges fast for matrices that are already close to triangular form, and thus it has the potential to become a useful tool for the solution of parameter-depending eigenvalue problems. Indeed, assume that $A(\omega)$ is a matrix-valued function that depends continuously on the parameter ω. Once the eigenvalue problem has solved for a specific value of ω, say, ω_0, then the transformation that reduces $A(\omega_0)$ to Schur form will transform matrices $A(\omega)$ to a form that is close to being triangular whenever ω is sufficiently close to ω_0. It may then be useful to apply a nonsymmetric Jacobi method to $A(\omega)$ to obtain the Schur form within one or two sweeps. Finally, it was shown in [7] that Jacobi's algorithm for symmetric matrices is more accurate than the QR algorithm if the right

stopping criterion is used. Extending this theory to nonsymmetric Jacobi methods may produce a highly accurate algorithm. In this paper, however, we restrict ourselves to the investigation of the asymptotic convergence behavior of the method.

The remainder of the paper is organized as follows. In the following section, we explain why the convergence theory for nonsymmetric Jacobi algorithms is challenging and different from the theory for the symmetric case. In section 3, we prove asymptotic quadratic convergence of the cyclic nonsymmetric Jacobi algorithm if so-called northeast directed sweeps are used. In section 4, we investigate generalizations of the algorithm to the solution of the Hamiltonian eigenvalue problem. In particular, we explain why the Jacobi-like algorithm proposed in [1] does not show asymptotic quadratic convergence and we show how convergence can be accelerated. Finally, we illustrate the theoretical results by the results of numerical experiments in section 5.

2. Why convergence of nonsymmetric Jacobi algorithms is not obvious. It is well known that Jacobi's classical algorithm as well as many cyclic versions are asymptotically quadratically convergent; see, e.g., [16, 17, 31, 35]. The same is also known for several generalizations; see [20] for a general proof of local quadratic convergence of Jacobi-type methods. However, these results are usually based on the minimization of a particular smooth function in each Jacobi step. For the standard eigenvalue problem with a Hermitian matrix $M = (m_{ij}) \in \mathbb{C}^{n\times n}$ this smooth function is the quantity

$$\operatorname{off}(M) := \sqrt{\sum_{i>j} |m_{ij}|^2} \tag{2.1}$$

that is sometimes called . In contrast to the Hermitian case, the quantity $\operatorname{off}(M)$ need not decrease in a single step of the nonsymmetric Jacobi algorithm. This effect can be explained heuristically with the help of the following sketch:

$$\begin{bmatrix} * & * & * & * & * & * \\ \cdot & \circ & * & * & \circ & * \\ \cdot & \diamond & * & * & \bullet & * \\ \cdot & \cdot & \cdot & * & * & * \\ \cdot & \circ & \cdot & \cdot & \circ & * \\ \cdot & \cdot & \cdot & \cdot & \cdot & * \end{bmatrix}$$

If the pivot element is chosen such that, currently, the 2×2 subproblem indicated by $\circ$ is under investigation, then a very large entry in, e.g., the position marked with $\bullet$ may lead to a temporary increase in $\operatorname{off}(M)$, because the element in the $\diamond$-position will be linearly combined with the element in the $\bullet$-position. It is this effect which makes the convergence analysis of nonsymmetric Jacobi methods so delicate. Indeed, the general results of [20] cannot be applied here, because $\operatorname{off}(M)$ is not minimized in each step.

At this stage, the reader might stop and ask whether it would not be advisable to modify the algorithm in such a way that $\operatorname{off}(M)$ decreases monotonically. Indeed, such generalizations have already been considered. For example, Stewart [32] proposed to use pivot elements only from the first subdiagonal. Indeed, this avoids the effect explained in the previous paragraph and it was shown that $\operatorname{off}(M)$ then decreases monotonically. But unfortunately, it turned out that the Jacobi algorithm obtained in this way is characterized by extreme slow convergence: it appears to be almost stagnant. On the other hand, one may also consider variants of nonsymmetric Jacobi algorithm that again allow pivot elements from the whole strict lower triangular part

of the matrix but that minimize $\mathrm{off}(M)$ in each step rather than annihilate the pivot element. This, however, would require the solution of a minimization problem in each step and one would have to take into account "global information," i.e., the knowledge of all elements in the strict lower triangular part of the matrix would be necessary. In contrast, the transformation annihilating the pivot element can be easily computed from considering the corresponding 2×2 problem only, thus only taking into account "local information." Consequently, a single Jacobi step of the modified method would be much more expensive than a single Jacobi step of a cyclic nonsymmetric Jacobi method. Thus, although $\mathrm{off}(M)$ does not decrease monotonically in each step, cyclic methods seem to be the cheapest and most reliable variants of nonsymmetric Jacobi algorithms for the general complex eigenvalue problem.

But the nonmonotone behavior of the quantity $\mathrm{off}(M)$ is not the only point in which the general case differs from the symmetric case. The asymptotic convergence behavior of a cyclic Jacobi method may be completely different for symmetric matrices on the one hand and general complex matrices on the other hand. As an example let us consider the cyclic Jacobi method using [illegible] given by the sequence of indices

$$\big((2,1),(3,1),\dots,(n,1),(3,2),\dots,(n,2),\dots,(n,n-1)\big) \tag{2.2}$$

versus the the one using [illegible] given by

$$\big((n,1),(n-1,1),\dots,(2,1),(n,2),\dots,(3,2),\dots,(n,n-1)\big). \tag{2.3}$$

Both methods appear to be asymptotically quadratically convergent in the symmetric case, while the nonsymmetric case shows quadratic convergence for the "bottom-to-top" variant but linear convergence for the "top-to-bottom" variant. (See section 5 for numerical experiments on this topic.) Again this effect can be explained heuristically with the help of a sketch:

$$\Downarrow \text{ top to bottom} \qquad \begin{bmatrix} \circ & * & * & \circ & * & * \\ \diamond & * & * & \bullet & * & * \\ \diamond & \cdot & * & \bullet & * & * \\ \circ & \cdot & \cdot & \circ & * & * \\ \cdot & \cdot & \cdot & \cdot & * & * \\ \cdot & \cdot & \cdot & \cdot & \cdot & * \end{bmatrix} \qquad \Uparrow \text{ bottom to top} \qquad \begin{bmatrix} \circ & * & * & \circ & * & * \\ \cdot & * & * & * & * & * \\ \cdot & \cdot & * & * & * & * \\ \circ & \cdot & \cdot & \circ & * & * \\ \diamond & \cdot & \cdot & \triangle & * & * \\ \diamond & \cdot & \cdot & \triangle & \cdot & * \end{bmatrix}$$

Assume that the matrix under consideration is already close to triangular form so that the algorithm may have reached the expected phase of quadratic convergence. If the pivot element is chosen in such a way that the corresponding 2×2 subproblem under consideration is the one displayed by the symbol $\circ$, then the elements marked by $\diamond$ are the ones that have already been annihilated once in the current sweep. If the top-to-bottom column-by-column sweep is used, then the current Jacobi step linearly combines those elements with possibly large elements from the strict upper triangular part from the matrix marked with the symbol $\bullet$. On the other hand, if the bottom-to-top column-by-column sweep is used, then the current Jacobi step linearly combines the $\diamond$-elements with elements from the strict lower triangular part, marked by the symbol $\triangle$, that are expected to be sufficiently small. Thus, the increase of modulus of elements that have already been annihilated once may be much higher when top-to-bottom column-by-column are used and that is exactly what can be observed in practice. In the symmetric case, however, this observation does not apply, because in this case also the elements in the strict upper triangular part are expected to be sufficiently small.

3. A proof of asymptotic quadratic convergence. Wilkinson's proof [35] of asymptotic quadratic convergence of the classical symmetric Jacobi method makes extensive use of the fact that the offnorm $\mathrm{off}(M)$ decreases monotonically over the steps of the algorithm. As pointed out in section 2, this is no longer true for the nonsymmetric Jacobi method. Thus, we will have to investigate in detail the possible changes in the moduli of the entries in the strict lower triangular part of the matrix in a single Jacobi step. Let us introduce the following notation. Starting with the matrix $M \in \mathbb{C}^{n\times n}$, let us denote by $M_\nu = (m_{ij}^{(\nu)})$ the matrix that we have obtained after performing ν Jacobi steps. (In particular, we have $M_0 = M$.) Moreover, we denote

$$\varrho_\nu := \min_{i\neq j} \left| m_{i,i}^{(\nu)} - m_{j,j}^{(\nu)} \right|, \tag{3.1}$$

$$\eta_\nu := \max\left\{ |m_{ij}^{(\nu)}| \;\middle|\; i,j = 1,\dots,n \right\}, \tag{3.2}$$

$$\varepsilon_\nu := \max\left\{ |m_{ij}^{(\nu)}| \;\middle|\; i > j \right\}, \tag{3.3}$$

i.e., ϱ_ν is the smallest distance between two diagonal elements of the matrix M_ν, η_ν is the modulus of the largest element in modulus of M_ν, and ε_ν is the modulus of the largest element in modulus of the strict lower triangular part of M_ν. In the following, let ν be fixed and assume we have

$$\varrho_\nu > 0 \quad \text{and} \quad 4\frac{\varepsilon_\nu \eta_\nu}{\varrho_\nu^2} < 1. \tag{3.4}$$

Suppose that the (k,l)-element of M_ν is the pivot element of the current $((\nu+1)\text{st})$ step of the algorithm. We then compute the unitary matrix

$$Q = \begin{bmatrix} \cos x & -e^{-i\alpha}\sin x \\ e^{i\alpha}\sin x & \cos x \end{bmatrix}, \quad x, \alpha \in \mathbb{R},$$

that satisfies

$$Q^* \begin{bmatrix} m_{ll}^{(\nu)} & m_{lk}^{(\nu)} \\ m_{kl}^{(\nu)} & m_{kk}^{(\nu)} \end{bmatrix} Q = \begin{bmatrix} m_{ll}^{(\nu+1)} & m_{lk}^{(\nu+1)} \\ 0 & m_{kk}^{(\nu+1)} \end{bmatrix} \tag{3.5}$$

and that is closest to the identity matrix among all matrices satisfying (3.5). To obtain an estimate for the modulus of $\sin x$, we will use the following lemma, which is a special case of Theorem V.2.1 in [33].

LEMMA 3.1. [illegible] $A \in \mathbb{C}^{2\times 2}$ [illegible]

$$A = \begin{bmatrix} a & \eta \\ \varepsilon & b \end{bmatrix}, \text{ [illegible]} \quad 4\frac{|\varepsilon|\cdot|\eta|}{|a-b|^2} < 1.$$

[illegible] v [illegible] A [illegible]

$$v = \begin{bmatrix} 1 \\ p \end{bmatrix} \text{ [illegible]} \quad |p| < 2\frac{\varepsilon}{|a-b|}.$$

Using this lemma and taking into account (3.4), it is clear that the parameter x in Q satisfies $|\sin x| < 2\varepsilon_\nu/\varrho_\nu$. If the transformation induced by Q is applied to M_ν,

then it acts only on elements in the kth and lth rows and columns of M_ν. For the elements that have been altered in $M_{\nu+1}$, we obtain that

$$\begin{aligned} m_{lj}^{(\nu+1)} &= m_{lj}^{(\nu)} \cos x + m_{kj}^{(\nu)} e^{-i\alpha} \sin x, \\ m_{kj}^{(\nu+1)} &= m_{kj}^{(\nu)} \cos x - m_{lj}^{(\nu)} e^{i\alpha} \sin x, \\ m_{il}^{(\nu+1)} &= m_{il}^{(\nu)} \cos x + m_{ik}^{(\nu)} e^{i\alpha} \sin x, \\ m_{ik}^{(\nu+1)} &= m_{ik}^{(\nu)} \cos x - m_{il}^{(\nu)} e^{-i\alpha} \sin x \end{aligned}$$

for $i, j = 1, \dots, n$. Using these identities and using $|\cos x| \leq 1$ and $|\sin x| < 2\varepsilon_\nu/\varrho_\nu$, we obtain that $|m_{ij}^{(\nu+1)}| \leq w_{ij}$, where w_{ij} is given in the following table:

(3.6)

w_{ij}	$j < l$	$j = l$	$l < j < k$	$j = k$	$k < j$
$i < l$	$\lvert m_{ij}^{(\nu)}\rvert$	$\eta_{\nu+1}$	$\lvert m_{ij}^{(\nu)}\rvert$	$\eta_{\nu+1}$	$\lvert m_{ij}^{(\nu)}\rvert$
$i = l$	$\lvert m_{ij}^{(\nu)}\rvert + 2\frac{\varepsilon_\nu^2}{\varrho_\nu}$	$\eta_{\nu+1}$	$\eta_{\nu+1}$	$\eta_{\nu+1}$	$\eta_{\nu+1}$
$l < i < k$	$\lvert m_{ij}^{(\nu)}\rvert$	$\lvert m_{ij}^{(\nu)}\rvert + 2\frac{\varepsilon_\nu \eta_\nu}{\varrho_\nu}$	$\lvert m_{ij}^{(\nu)}\rvert$	$\eta_{\nu+1}$	$\lvert m_{ij}^{(\nu)}\rvert$
$i = k$	$\lvert m_{ij}^{(\nu)}\rvert + 2\frac{\varepsilon_\nu^2}{\varrho_\nu}$	0	$\lvert m_{ij}^{(\nu)}\rvert + 2\frac{\varepsilon_\nu \eta_\nu}{\varrho_\nu}$	$\eta_{\nu+1}$	$\eta_{\nu+1}$
$k < i$	$\lvert m_{ij}^{(\nu)}\rvert$	$\lvert m_{ij}^{(\nu)}\rvert + 2\frac{\varepsilon_\nu^2}{\varrho_\nu}$	$\lvert m_{ij}^{(\nu)}\rvert$	$\lvert m_{ij}^{(\nu)}\rvert + 2\frac{\varepsilon_\nu^2}{\varrho_\nu}$	$\lvert m_{ij}^{(\nu)}\rvert$

In this table, we can see the effect that has been heuristically explained in section 2. If we are in the stage that $\varepsilon_\nu \ll \eta_\nu$, i.e., our matrix under consideration is already close to triangular form, then the modulus of entries in the strict lower triangular part in positions (i, j) with $j < l$ or $k < i$ may have increased by $2\varepsilon_\nu^2/\varrho_\nu$ only, while the modulus of entries in positions (i, j) with $l \leq j$ and $i \leq k$ may have increased by $2\eta_\nu\varepsilon_\nu/\varrho_\nu \gg 2\varepsilon_\nu^2/\varrho_\nu$. Thus, the sweep should start from the lower left corner of the lower triangular part and then proceed to the northeast in order to guarantee that entries that have been eliminated once are increased by at most $2\varepsilon_\nu^2/\varrho_\nu$. This motivates the following definition.

DEFINITION 3.2. [illegible] $n \in \mathbb{N}$, $N = \frac{n(n-1)}{2}$ [illegible] $S = \big((i_1, j_1), (i_2, j_2) \dots, (i_N, j_N)\big)$ [illegible] N [illegible] $(i_\nu, j_\nu) \in \{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$, [illegible] $i_\nu > j_\nu$, $\nu = 1, \dots, N$ [illegible] S [illegible] northeast directed sweep sequence

$$\nu < \mu \quad \Rightarrow \quad (i_\nu > i_\mu, \quad j_\nu < j_\mu)$$

[illegible] $\nu, \mu \in \{1, \dots, N\}$ [illegible] northeast directed sweep

Particular example of northeast directed sweep sequences are given by the sequence (2.3) that induces a bottom-to-top column-by-column sweep or by the sequence

$$\big((n, 1), (n-1, 1), (n, 2), (n-2, 1), (n-1, 2), (n, 3), \dots, (2, 1), (3, 2), \dots, (n, n-1)\big).$$

We continue by analyzing how the values of ε_ν, η_ν, and ϱ_ν have changed after one Jacobi step (regardless what kind of sweep is used). Since $\varepsilon_\nu \leq \eta_\nu$, we obtain from the discussion above that

$$\varepsilon_{\nu+1} \leq \varepsilon_\nu \left(1 + 2\frac{\eta_\nu}{\varrho_\nu}\right). \tag{3.7}$$

We could also produce a bound for $\eta_{\nu+1}$, but for our purpose it is sufficient to note that η_μ is bounded by $\|M\|_F$ for any $\mu \in \mathbb{N} \cup \{0\}$. It remains to investigate how ϱ_ν has changed. Therefore, we have to investigate in particular the changes on the diagonal of M_ν. Clearly, we have

$$\begin{aligned} m_{ll}^{(\nu+1)} &= m_{ll}^{(\nu)}(1-\sin^2 x) + m_{kl}^{(\nu)} e^{-ia}\sin x\cos x + m_{lk}^{(\nu)} e^{ia}\sin x\cos x + m_{kk}^{(\nu)}\sin^2 x, \\ m_{kk}^{(\nu+1)} &= m_{ll}^{(\nu)}\sin^2 x - m_{kl}^{(\nu)} e^{-ia}\sin x\cos x - m_{lk}^{(\nu)} e^{ia}\sin x\cos x + m_{kk}^{(\nu)}(1-\sin^2)x. \end{aligned}$$

Thus, we obtain $m_{ll}^{(\nu+1)} = m_{ll}^{(\nu)} + \Delta_l$, $m_{kk}^{(\nu+1)} = m_{kk}^{(\nu)} + \Delta_k$, where

$$\begin{aligned} &|\Delta_l| = |\Delta_k| \\ &\le \left|m_{ll}^{(\nu)}\sin^2 x\right| + \left|m_{kl}^{(\nu)} e^{-ia}\sin x\cos x\right| + \left|m_{lk}^{(\nu)} e^{ia}\sin x\cos x\right| + \left|m_{kk}^{(\nu)}\sin^2 x\right| \\ &\le \eta_\nu \cdot 4\frac{\varepsilon_\nu^2}{\varrho_\nu^2} + \varepsilon_\nu \cdot 2\frac{\varepsilon_\nu}{\varrho_\nu} + \eta_\nu \cdot 2\frac{\varepsilon_\nu}{\varrho_\nu} + \eta_\nu \cdot 4\frac{\varepsilon_\nu^2}{\varrho_\nu^2} \\ &\le 2\left(\frac{\varepsilon_\nu^2}{\varrho_\nu} + 4\eta_\nu\frac{\varepsilon_\nu^2}{\varrho_\nu^2} + \eta_\nu\frac{\varepsilon_\nu}{\varrho_\nu}\right). \end{aligned}$$

It follows that $|m_{ii}^{(\nu+1)} - m_{jj}^{(\nu+1)}| \ge |m_{ii}^{(\nu)} - m_{jj}^{(\nu)}| - |\Delta_l| - |\Delta_k|$ for all $i, j = 1, \dots, n$, and thus

$$\varrho_{\nu+1} \ge \varrho_\nu - 4\left(\frac{\varepsilon_\nu^2}{\varrho_\nu} + 4\eta_\nu\frac{\varepsilon_\nu^2}{\varrho_\nu^2} + \eta_\nu\frac{\varepsilon_\nu}{\varrho_\nu}\right). \tag{3.8}$$

Using the above, we will now show that the cyclic nonsymmetric Jacobi method using northeast directed sweeps is quadratically convergent if the matrix under consideration is sufficiently close to triangular form.

THEOREM 3.3. [illegible] $M = (m_{ij}) \in \mathbb{C}^{n\times n}$ [illegible] $M_\nu = (m_{ij}^{(\nu)})$ [illegible] M [illegible] ν [illegible] δ_0 [illegible] δ_μ [illegible] $M = M_0$ [illegible] M [illegible] μ [illegible]

$$\delta_\mu := \max\left\{|m_{ij}^{(\mu N)}| \;\middle|\; i > j\right\}, \quad \mu \in \mathbb{N} \cup \{0\}. \tag{3.9}$$

[illegible] $N := n(n-1)/2$ [illegible]

$$\eta := \|M\|_F, \quad \varrho := \frac{1}{2}\min_{i\ne j}|m_{ii} - m_{jj}|, \quad \delta := 2\delta_0\left(1 + 2\frac{\eta}{\varrho}\right)^N. \tag{3.10}$$

[illegible] $\varrho > 0$ [illegible] δ_0 [illegible]

$$\frac{\delta\eta}{\varrho^2} < \frac{1}{4}, \qquad \frac{\delta^2}{\varrho} + 4\eta\frac{\delta^2}{\varrho^2} + \eta\frac{\delta}{\varrho} \le \frac{\varrho_0}{4N}, \qquad 2\frac{N\delta^2}{\varrho} \le \delta_0, \tag{3.11}$$

[illegible] $\mu \in \mathbb{N} \cup \{0\}$ [illegible]

$$\delta_{(\mu+1)N} \le \left(1 + 2\frac{\eta}{\varrho}\right)^{2N}\frac{2N}{\varrho}\delta_{\mu N}^2,$$

[illegible]

. Let η_ν, ε_ν, and ϱ_ν be defined as in (3.1)–(3.3). Then we have

$$\delta_\mu = \varepsilon_{\mu N} \quad \text{and} \quad \varrho = \frac{\varrho_0}{2}.$$

From (3.7) and (3.8) we obtain that ε_ν (and ϱ_ν, respectively) may increase (or decrease, respectively) in each Jacobi step. We first show by induction that this increase (decrease, respectively) remains under control, i.e., that for $\mu \in \mathbb{N} \cup \{0\}$ and $p = 0, \dots, N$, we have that

$$\varepsilon_{\mu N} \le \frac{\delta_0}{2^\mu}, \qquad \varrho_{\mu N} \ge \varrho_0 - \sum_{j=1}^{\mu} \frac{\varrho_0}{2^{j+1}} > \varrho, \quad \text{and} \tag{3.12}$$

$$\varepsilon_{\mu N+p} \le \varepsilon_{\mu N} \left(1 + 2\frac{\eta}{\varrho}\right)^p, \qquad \varrho_{\mu N+p} \ge \varrho_{\mu N} - p\frac{\varrho_0}{2^{\mu+1}N}. \tag{3.13}$$

$(\mu, p) = (0, 0)$: There is nothing to prove.

$(\mu, p) \Rightarrow (\mu, p+1)$: Let $p \le N$. By the induction hypothesis for (μ, p) and $(\mu, 0)$, we have that

$$\varepsilon_{\mu N+p} \le \varepsilon_{\mu N} \left(1 + 2\frac{\eta}{\varrho}\right)^p \le \frac{\delta_0}{2^\mu} \left(1 + 2\frac{\eta}{\varrho}\right)^p \le \frac{\delta}{2^{\mu+1}} \tag{3.14}$$

using (3.10) and

$$\varrho_{\mu N+p} \ge \varrho_{\mu N} - p\frac{\varrho_0}{2^{\mu+1}N} \ge \varrho_0 - \sum_{j=1}^{\mu} \frac{\varrho_0}{2^{j+1}} - N\frac{\varrho_0}{2^{\mu+1}N} = \varrho_0 - \sum_{j=1}^{\mu+1} \frac{\varrho_0}{2^{j+1}} > \varrho. \tag{3.15}$$

Now assume $p < N$. Then we obtain using (3.7), (3.8), (3.14), and (3.15) and the induction hypothesis that

$$\begin{aligned}
\varepsilon_{\mu N+p+1} &\le \varepsilon_{\mu N+p} \left(1 + 2\frac{\eta_{\mu N+p}}{\varrho_{\mu N+p}}\right) \le \varepsilon_{\mu N+p} \left(1 + 2\frac{\eta}{\varrho}\right) \le \varepsilon_{\mu N} \left(1 + 2\frac{\eta}{\varrho}\right)^{p+1}; \\
\varrho_{\mu N+p+1} &\ge \varrho_{\mu N+p} - 4\left(\frac{\varepsilon_{\mu N+p}^2}{\varrho_{\mu N+p}} + 4\eta_{\mu N+p}\frac{\varepsilon_{\mu N+p}^2}{\varrho_{\mu N+p}^2} + \eta_{\mu N+p}\frac{\varepsilon_{\mu N+p}}{\varrho_{\mu N+p}}\right) \\
&\ge \varrho_{\mu N} - p\frac{\varrho_0}{2^{\mu+1}N} - 4\left(\frac{\delta^2}{(2^{\mu+1})^2\varrho} + 4\eta\frac{\delta^2}{(2^{\mu+1})^2\varrho^2} + \eta\frac{\delta}{2^{\mu+1}\varrho}\right) \\
&\ge \varrho_{\mu N} - p\frac{\varrho_0}{2^{\mu+1}N} - \frac{4}{2^{\mu+1}}\left(\frac{\delta^2}{\varrho} + 4\eta\frac{\delta^2}{\varrho^2} + \eta\frac{\delta}{\varrho}\right) \\
&\ge \varrho_{\mu N} - (p+1)\frac{\varrho_0}{2^{\mu+1}N}. \qquad \Big(\text{by (3.11)}\Big).
\end{aligned}$$

$(\mu, p) \Rightarrow (\mu+1, 0)$: For obtaining a bound for $\varepsilon_{(\mu+1)N}$, let us note that during the $(\mu+1)$st sweep each entry (i, j) in the strict lower triangular part of the current pencil is set to zero at one step. Afterward, during the remainder of the sweep, it is affected $k < N$ times in the steps, say, $\mu N + \ell_1, \dots, \mu N + \ell_k$, where k and $\ell_1, \dots, \ell_k$ depend on i, j. Since a northeast directed sweep is used, we obtain from table (3.6) that the modulus of the (i, j)-element of the matrix $M_{((\mu+1)N)}$ obtained after completing the

$(\mu+1)$st sweep is bounded by

$$|m_{ij}^{((\mu+1)N)}| \le 2\frac{\varepsilon_{\mu N+\ell_1}^2}{\varrho_{\mu N+\ell_1}} + \cdots + 2\frac{\varepsilon_{\mu N+\ell_k}^2}{\varrho_{\mu N+\ell_k}} \le \sum_{p=1}^{N} 2\frac{\varepsilon_{\mu N+p}^2}{\varrho_{\mu N+p}} \le N\left(\frac{\delta}{2^{\mu+1}}\right)^2 \frac{2}{\varrho}, \tag{3.16}$$

because $k \le N$ and $\varepsilon_{\mu N+p} \le \delta/2^{\mu+1}$ and $\varrho_{\mu N+p} \ge \varrho$ for $p = 0, \ldots, N$. Since the right-hand side of (3.16) is independent of the indices i and j, we then obtain

$$\varepsilon_{(\mu+1)N} \le N\left(\frac{\delta}{2^{\mu+1}}\right)^2 \frac{2}{\varrho} \le \frac{\delta_0}{(2^{\mu+1})^2} \le \frac{\delta_0}{2^{\mu+2}}. \tag{3.17}$$

This concludes the proof of (3.12) and (3.13). Now observe that (3.13) implies that

$$\varepsilon_{\mu N+p} \le \varepsilon_{\mu N}\left(1+2\frac{\eta}{\varrho}\right)^N = \delta_\mu\left(1+2\frac{\eta}{\varrho}\right)^N$$

for $p = 0, \ldots, N$. Using this inequality instead of (3.14), we obtain analogously to (3.17) that

$$\delta_{\mu+1} = \varepsilon_{(\mu+1)N} \le \sum_{p=1}^{N} 2\frac{\varepsilon_{\mu N+p}^2}{\varrho_{\mu N+p}} \le N\delta_\mu^2\left(1+2\frac{\eta}{\varrho}\right)^{2N}\frac{2}{\varrho},$$

which concludes the proof. □

Theorem 3.3 guarantees asymptotic quadratic convergence in terms of the largest modulus ε_ν of subdiagonal entries. Since the so-called offnorm

$$\operatorname{off}(M_\nu) := \sqrt{\sum_{i>j} |m_{ij}^{(\nu)}|^2}$$

is bounded by $\varepsilon_\nu \le \operatorname{off}(M_\nu) \le \sqrt{N}\varepsilon_\nu$, we also obtain asymptotic quadratic convergence in terms of the offnorm.

3.4. Note that in general the assumption $\varrho > 0$ in Theorem 3.3 cannot be weakened in order to guarantee convergence. Consider, for example, the matrix

$$M = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ \varepsilon & 0 & 1 \end{bmatrix},$$

where $\varepsilon > 0$ is arbitrarily small. Then the sequence of matrices generated by the nonsymmetric Jacobi method becomes periodic as long as exact arithmetic is performed:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ \varepsilon & 0 & 1 \end{bmatrix} \rightsquigarrow \begin{bmatrix} 1 & 0 & \varepsilon \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \rightsquigarrow \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & \varepsilon \\ 1 & 0 & 1 \end{bmatrix} \rightsquigarrow \begin{bmatrix} 1 & 0 & 1 \\ \varepsilon & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$
$$\rightsquigarrow \begin{bmatrix} 1 & \varepsilon & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \rightsquigarrow \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & \varepsilon & 1 \end{bmatrix} \rightsquigarrow \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ \varepsilon & 0 & 1 \end{bmatrix}.$$

However, if finite precision arithmetic is used, then roundoff errors break the periodicity and the algorithm starts to converge. For example, with $\varepsilon = 1/100$ our Matlab implementation of the algorithm needed eight sweeps for convergence.

4. Hamiltonian Jacobi methods. As pointed out in the introduction, the nonsymmetric Jacobi method may not be competitive in comparison to the highly efficient QR algorithm. However, this changes if one is interested in structure-preserving algorithms for the solution of structured eigenvalue problems. As an example, we consider the [illegible]. A matrix $H \in \mathbb{C}^{2n\times 2n}$ is called [illegible] if

$$H^T J + JH = 0, \quad \text{where } J = J_{2n} = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix},$$

or, equivalently, if

$$H = (h_{ij}) = \begin{bmatrix} A & C \\ D & -A^T \end{bmatrix}, \quad \text{where } A, C, D \in \mathbb{C}^{n\times n},\ C = C^T,\ D = D^T. \tag{4.1}$$

In some sources, matrices $H \in \mathbb{C}^{2n\times 2n}$ satisfying $H^*J + JH = 0$ are called Hamiltonian. For the sake of clear distinction, we refer to such matrices as [illegible], while we call matrices satisfying (4.1) [illegible]. The real and complex conjugate Hamiltonian eigenvalue problems have been extensively studied in the literature; see, e.g., [22, 28] and the references therein. Also, the complex transpose Hamiltonian eigenvalue problem has attracted some attention in recent years due to its relation to the so-called palindromic eigenvalue problem that arises in an application in the vibration analysis of rail tracks; see [18, 26, 25].

The solution of the Hamiltonian eigenvalue problem is in general tackled by computing condensed forms under unitary symplectic similarity transformations, i.e., transformations of the form $H \mapsto U^{-1}HU$, where $U \in \mathbb{F}^{2n\times 2n}$ is unitary and [illegible], i.e., it satisfies $U^\star JU = J$, where $\star = T$ in the real Hamiltonian and complex transpose Hamiltonian case and $\star = *$ in the complex conjugate Hamiltonian case. It is easy to check that unitary symplectic similarity transformations preserve the Hamiltonian structure, i.e., if H is Hamiltonian, then so is $U^{-1}HU$. The condensed form one is aiming at is the [illegible]. A Hamiltonian matrix H is said to be in Hamiltonian Schur form if

$$H = \begin{bmatrix} R & B \\ 0 & -R^\star \end{bmatrix}, \tag{4.2}$$

where $\star = T$ or $\star = *$, respectively. This form can always be achieved for complex transpose Hamiltonian matrices. For complex conjugate Hamiltonian matrices, it can be achieved if there are no eigenvalues on the imaginary axis [23].

What makes the Hamiltonian eigenvalue problem challenging is the fact that although a Hamiltonian QR algorithm, i.e., a structure-preserving version of the QR algorithm, has been developed [2], a preliminary structure-preserving reduction to a Hessenberg-like form is missing—a phenomenon that is known in the literature as Van Loan's curse [6, 30]. Thus, the Hamiltonian QR algorithm is in general not efficient, because it is of complexity $\mathcal{O}(n^4)$, and so, the competition is open for other kinds of structure-preserving algorithms.

In [3] and [1] Jacobi-like algorithms for Hamiltonian matrices have been proposed. Both algorithms are based on the solution of 4×4 subproblems in each Jacobi step. Byers [3] follows the idea of Stewart [32] of using only a selected set of pivot elements which results in an extreme slow convergence behavior. Therefore, we omit a detailed discussion of the first algorithm and focus on the algorithm proposed in [1] as well

as on a direct generalization of the nonsymmetric Jacobi method to Hamiltonian matrices which results in a Hamiltonian Jacobi method that is based on the solution of 2×2 subproblems in each step.

We restrict our attention to complex transpose Hamiltonian matrices. (A generalization of the discussion to the case of complex conjugate Hamiltonian matrices is possible, but more involved, because one has to take into account the fact that some of the subproblems may not be solvable. This is because reduction to Hamiltonian Schur form is not always possible in the complex conjugate Hamiltonian case if there are eigenvalues on the imaginary axis.) Our transformation matrices are supposed to be unitary and symplectic, so they must have the form

$$U = \begin{bmatrix} U_1 & -\overline{U}_2 \\ U_2 & \overline{U}_1 \end{bmatrix}, \quad U_1, U_2 \in \mathbb{C}^{n \times n},$$

where $U_1^* U_1 + U_2^* U_2 = I_n$ and $U_2^T U_1 - U_1^T U_2 = 0$.

The reason 4×4 subproblems instead of 2×2 problems are considered in the algorithm proposed in [1] becomes obvious from the following sketch:

$$\left[\begin{array}{cccc|cccc} * & * & * & * & * & * & * & * \\ \cdot & \bullet & \bullet & * & * & \bullet & \bullet & * \\ \cdot & \diamond & \bullet & * & * & \bullet & \bullet & * \\ \cdot & \cdot & \cdot & * & * & * & * & * \\ \hline \cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\ \cdot & \circ & \circ & \cdot & * & \bullet & \circ & \cdot \\ \cdot & \circ & \circ & \cdot & * & \bullet & \bullet & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & * & * & * \end{array}\right]$$

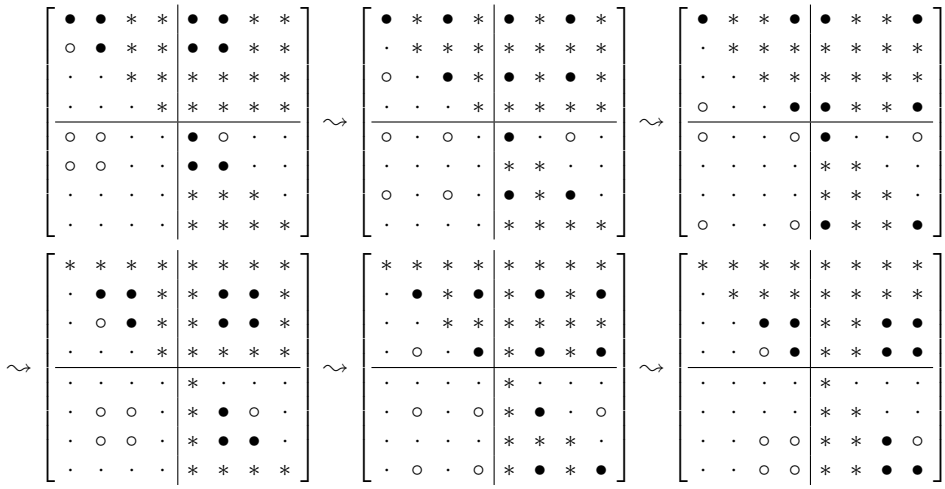
If the element displayed by $\diamond$ has been chosen as pivot element, then the 4×4 submatrix displayed by the symbols $\circ$ and $\bullet$ is the smallest Hamiltonian submatrix that contains the pivot element. (Here, $\circ$ refers to elements in the subproblem that are annihilated, while $\bullet$ stands for entries that may remain large in norm.) For details concerning the solution of the 4×4 subproblem, see [1]. (The discussion in [1] involves complex conjugate Hamiltonian matrices only, but the generalization to the complex transpose Hamiltonian case is straightforward.) Among all possible transformation matrices, we once again choose the one that is closest to the identity in order to enable asymptotic quadratic convergence.

A sweep of the Hamiltonian Jacobi algorithm as proposed in [1] is then given by the following sequence of indices, where the quadrupel of indices (i, j, k, l) refers to the 4×4 subproblem consisting of the rows and columns i, j, k, l:

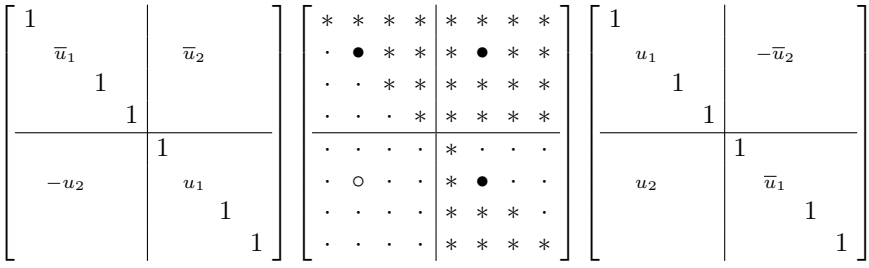
$$\begin{aligned} &(1, 2, n+1, n+2), (1, 3, n+1, n+3), \ldots, (1, n, n+1, 2n), \\ &(2, 3, n+2, n+3), (2, 4, n+2, n+4), \ldots, (n-1, n, 2n-1, 2n). \end{aligned} \tag{4.3}$$

Indeed, one easily checks that each pivot element is eliminated at least once during the sweep. On the other hand, the elements in the $(j, n+j)$ positions are annihilated $n-1$ times in each sweep, a fact that cannot be avoided when Hamiltonian 4×4 subproblems are considered. In the following, we will refer to the cyclic Hamiltonian Jacobi algorithm as proposed in [1] as [illegible] when sweeps based on the sequence of indices (4.3) are used. As an example, we display such a sweep for an

8×8 Hamiltonian matrix in the sketch below:



Next we consider a Jacobi-like algorithm for Hamiltonian matrices that is based on the solution of 2×2 subproblems. (This algorithm is related to the algorithm JIGH2 in [27] that was designed for the solution of the generalized Hermitian eigenvalue problem.) In this algorithm, we have to distinguish between pivot elements that are on the diagonal of C in (4.1) and those that are not. Indeed, if we select a pivot element $h_{n+k,k}$ from the diagonal of C, then there is a corresponding Hamiltonian 2×2 subproblem that contains this element, as indicated in the following sketch:



This 2×2 subproblem can be solved as in the nonsymmetric Jacobi method for general complex matrices by choosing $x, \alpha \in \mathbb{R}$ such that

$$Q = \begin{bmatrix} u_1 & -\overline{u}_2 \\ u_2 & \overline{u}_1 \end{bmatrix} = \begin{bmatrix} \cos x & -e^{-i\alpha} \sin x \\ e^{i\alpha} \sin x & \cos x \end{bmatrix}$$

annihilates the pivot element $h_{n+k,k}$ and is the matrix that is closest to the identity among all matrices that do so. If the matrix U is then obtained from the $2n \times 2n$ identity matrix by substituting $u_{kk} = u_{n+k,n+k} = u_1$, and $u_{k,n+k} = -\overline{u}_2$, $u_{n+k,k} = u_2$, we find that U is not only unitary, but also symplectic and thus the transformation with U will preserve the Hamiltonian structure of H.

The situation is different when the pivot element is not on the diagonal of C as then there is no Hamiltonian 2×2 subproblem that contains the given pivot element. To be specific, let $h_{k\ell}$ be a pivot element satisfying $\ell < k \leq n$ or $n + \ell < k \leq 2n$, i.e., the pivot element is in the strict lower triangular part of either the block A or the

block C in (4.1). As in the previous we then compute the matrix

$$Q = \begin{bmatrix} u_1 & -\overline{u}_2 \\ u_2 & \overline{u}_1 \end{bmatrix} = \begin{bmatrix} \cos x & -e^{-i\alpha}\sin x \\ e^{i\alpha}\sin x & \cos x \end{bmatrix} \tag{4.4}$$

that triangularizes the 2×2 submatrix $\hat{H}$ consisting of the elements $h_{\ell\ell}$, $h_{\ell k}$, $h_{k\ell}$, and h_{kk} and that is closest to the identity among all matrices of the form (4.4) that do so. Note that Q is not only unitary but also symplectic. However, when embedding Q into an $2n\times 2n$ matrix U by setting U to be the identity matrix except for the elements $u_{\ell\ell} = u_1$, $u_{\ell k} = -\overline{u}_2$, $u_{k\ell} = u_2$, $u_{kk} = \overline{u}_1$, then the resulting matrix U is unitary, but not symplectic, so we have to set $u_{n+\ell,n+\ell} = \overline{u}_1$, $u_{n+\ell,p} = -u_2$, $u_{p,n+\ell} = \overline{u}_2$, and $u_{p,p} = u_1$, where $p = n+k$ if $k \le n$ and $p = k-n$ if $n+\ell < k \le 2n$ to make it unitary and symplectic. Let us investigate this in detail. Denote $A = (a_{ij})$, $C = (c_{ij})$, $D = (d_{ij})$, where the submatrices A, C, D are given as in (4.1), and consider first the case $k \le n$, i.e., we have

$$\hat{H} = \begin{bmatrix} a_{\ell\ell} & a_{\ell k} \\ a_{k\ell} & a_{kk} \end{bmatrix},$$

and thus the pivot element is from the strict lower triangular part of A. The situation is depicted in the following sketch, where the pivot element is displayed with the symbol $\circ$ and the subproblem $\hat{H}$ is marked with the symbols $\circ$ and $\bullet$:

$$\left[\begin{array}{cccc|cccc} 1 & & & & & & & \\ & \overline{u}_1 & \overline{u}_2 & & & & & \\ & -u_2 & u_1 & & & & & \\ & & & 1 & & & & \\ \hline & & & & 1 & & & \\ & & & & & u_1 & u_2 & \\ & & & & & -\overline{u}_2 & \overline{u}_1 & \\ & & & & & & & 1 \end{array}\right] \left[\begin{array}{cccc|cccc} * & * & * & * & * & * & * & * \\ \cdot & \bullet & \bullet & * & * & * & * & * \\ \cdot & \circ & \bullet & * & * & * & * & * \\ \cdot & \cdot & \cdot & * & * & * & * & * \\ \hline \cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & + & \diamond & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & + & + & \cdot \\ \cdot & \cdot & \cdot & \cdot & * & * & * & * \end{array}\right] \left[\begin{array}{cccc|cccc} 1 & & & & & & & \\ & u_1 & -\overline{u}_2 & & & & & \\ & u_2 & \overline{u}_1 & & & & & \\ & & & 1 & & & & \\ \hline & & & & 1 & & & \\ & & & & & \overline{u}_1 & -u_2 & \\ & & & & & \overline{u}_2 & u_1 & \\ & & & & & & & 1 \end{array}\right]$$

Due to the special structure of U, we find that a second 2×2 subproblem given by

$$\widetilde{H} = \begin{bmatrix} h_{n+\ell,n+\ell} & h_{n+\ell,n+k} \\ h_{n+k,n+\ell} & h_{n+k,n+k} \end{bmatrix} = \begin{bmatrix} -a_{\ell\ell} & -a_{k\ell} \\ -a_{\ell k} & -a_{kk} \end{bmatrix} = -\hat{H}^T$$

is solved as well. This subproblem is depicted in the sketch above by the symbols $+$ and $\diamond$, where the symbol $\diamond$ display the element that will be annihilated by the transformation with U. Indeed, $\widetilde{H}$ will be transformed as

$$\begin{bmatrix} u_1 & u_2 \\ -\overline{u}_2 & \overline{u}_1 \end{bmatrix} \widetilde{H} \begin{bmatrix} \overline{u}_1 & -u_2 \\ \overline{u}_2 & u_1 \end{bmatrix} = -Q^T \hat{H}^T \overline{Q} = -(Q^* \hat{H} Q)^T = \begin{bmatrix} * & 0 \\ * & * \end{bmatrix}.$$

We have a similar situation for the case $n+\ell < k \le 2n$, i.e., when the pivot element is from the strict lower triangular part of C. Again, besides

$$\hat{H} = \begin{bmatrix} a_{\ell\ell} & c_{\ell k} \\ d_{k\ell} & -a_{kk} \end{bmatrix},$$

a second subproblem $\widetilde{H}$ is solved when the transformation with U is applied. Here, we have

$$\widetilde{H} = \begin{bmatrix} h_{k-n,k-n} & h_{k-n,n+\ell} \\ h_{n+\ell,k-n} & h_{n+\ell,n+\ell} \end{bmatrix} = \begin{bmatrix} a_{kk} & c_{\ell k} \\ d_{k\ell} & -a_{\ell\ell} \end{bmatrix} = -J_2^T \hat{H}^T J_2, \quad J_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

since $c_{k\ell} = c_{\ell k}$ and $d_{k\ell} = d_{\ell k}$. Then, $\widetilde{H}$ will be transformed as

$$Q^*\widetilde{H}Q = (J_2^T Q^T J_2)(-J_2^T \hat{H}^T J_2)(J_2^T \overline{Q} J_2) = -J_2^T (Q^* \hat{H} Q)^T J_2 = \begin{bmatrix} * & * \\ 0 & * \end{bmatrix},$$

where we used that Q is both unitary and symplectic, i.e., $Q = J_2^T Q^{-T} J_2 = J_2^T \overline{Q} J_2$. The situation is depicted in the sketch below, where $\hat{H}$ is displayed by the symbols $\circ$ and $\bullet$ and $\widetilde{H}$ is displayed by the symbols $\diamond$ and $+$:

$$\left[\begin{array}{ccc|ccc}
1 & & & & & \\
 & \overline{u}_1 & & & \overline{u}_2 & \\
 & & \overline{u}_1 & & \overline{u}_2 & \\
 & & & 1 & & \\
\hline
 & & & 1 & & \\
 & & -u_2 & & u_1 & \\
 & -u_2 & & & & u_1 \\
 & & & & & 1
\end{array}\right]
\left[\begin{array}{cccc|cccc}
* & * & * & * & * & * & * & * \\
\cdot & \bullet & * & * & * & * & \bullet & * \\
\cdot & \cdot & + & * & * & + & * & * \\
\cdot & \cdot & \cdot & * & * & * & * & * \\
\hline
\cdot & \cdot & \cdot & \cdot & * & \cdot & \cdot & \cdot \\
\cdot & \cdot & \diamond & \cdot & * & + & \cdot & \cdot \\
\cdot & \circ & \cdot & \cdot & * & * & \bullet & \cdot \\
\cdot & \cdot & \cdot & \cdot & * & * & * & *
\end{array}\right]
\left[\begin{array}{ccc|ccc}
1 & & & & & \\
 & u_1 & & & -\overline{u}_2 & \\
 & & u_1 & & -\overline{u}_2 & \\
 & & & 1 & & \\
\hline
 & & & 1 & & \\
 & & u_2 & & \overline{u}_1 & \\
 & u_2 & & & & \overline{u}_1 \\
 & & & & & 1
\end{array}\right]$$

(Again, $\circ$ and $\diamond$ stand for the elements that are annihilated by the transformation with U.)

It becomes clear from this discussion that it is sufficient to annihilate all elements in the strict lower triangular parts of A and C during one sweep. But how should a sweep be organized? The answer follows by noting that the Hamiltonian matrix H as in (4.2) is in Hamiltonian Schur form if and only if the matrix

$$\mathcal{F}H\mathcal{F} = \begin{bmatrix} R & BF_n \\ 0 & -F_n R^T F_n \end{bmatrix}, \quad \mathcal{F} = \begin{bmatrix} I_n & 0 \\ 0 & F_n \end{bmatrix}, \quad F_n = \begin{bmatrix} 0 & & 1 \\ & \iddots & \\ 1 & & 0 \end{bmatrix} \tag{4.5}$$

is in Schur form. Here, F_n denotes the [illegible], i.e., the matrix with ones on the southeast northwest diagonal and zeros elsewhere. (It is straightforward to check that a matrix L is lower triangular if and only if $F_n L F_n$ is upper triangular.) If we then carry out a [illegible] given by the sequence of indices

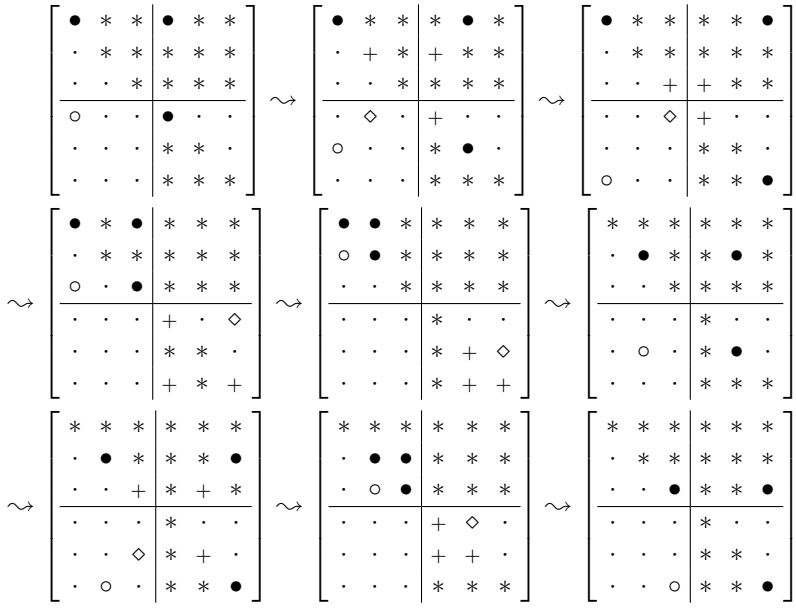
$$((n{+}1,1),(n{+}2,1),\ldots,(2n,1),(n,1),(n{-}1,1),\ldots,(2,1),(n{+}2,2),(n{+}3,2),\ldots,(2n,n)),$$

then given the fact that most of the time two elements are annihilated during a single Jacobi step, we find that the elements are annihilated in the order

$$\begin{aligned}
&((n{+}1,1),(n{+}2,1),(n{+}1,2),(n{+}3,1),(n{+}1,3),\ldots,(2n,1),(n{+}1,n),\\
&(n,1),(n{+}1,2n),(n{-}1,1),(n{+}1,2n{-}1),\ldots,(2,1),(n{+}1,n{+}2),\\
&(n{+}2,2),(n{+}3,2),(n{+}2,3),\ldots,(2n,n)).
\end{aligned}$$

It is then straightforward to check that this corresponds to a northeast directed sweep on the matrix $\mathcal{F}H\mathcal{F}$, and thus the Hamiltonian Jacobi algorithm for H using a Hamiltonian sweep corresponds to the nonsymmetric Jacobi algorithm for $\mathcal{F}H\mathcal{F}$ using a special northeast directed sweep. Therefore, by Theorem 3.3 we expect asymptotic quadratic convergence for the Hamiltonian Jacobi algorithm when Hamiltonian sweeps are used, and this is exactly what can be observed in numerical experiments. We de-

pict one Hamiltonian sweep for the case $n = 3$:



The Hamiltonian Jacobi algorithm based on the solution of 2×2 subproblems will be referred to as [illegible] 2 [illegible] if Hamiltonian sweeps are used.

Let us return to the Hamiltonian 4-Jacobi. Surprisingly, only linear asymptotic convergence of the method can be observed, as we will see in section 5. This behavior can be explained by looking at the order in which the pivot elements are annihilated. One easily finds that, for example, the pivot element in the $(2,1)$-position is annihilated in the step preceding the annihilation of the element in the $(3,1)$-position. Thus, the sweep given by the sequence of indices (4.3) [illegible] on the matrix $\mathcal{F}H\mathcal{F}$. In order to ensure this, we have to modify the sequence of indices as

$$\begin{aligned}
&(1,2,n{+}1,n{+}2),(1,3,n{+}1,n{+}3),\dots,(1,n{-}1,n{+}1,2n{-}1),(1,n,n{+}1,2n),\\
&(1,n{-}1,n{+}1,2n{-}1),\dots,(1,3,n{+}1,n{+}3),(1,2,n{+}1,n{+}2),\\
&(2,3,n{+}2,n{+}3),\dots,(2,n,n{+}2,2n),\dots,(2,3,n{+}2,n{+}3),\dots,(n{-}1,n,2n{-}1,2n).
\end{aligned} \tag{4.6}$$

Here, the majority of pivot elements is annihilated at least twice during a sweep, but the order in which the pivot elements are annihilated the last time during a sweep is now

$$((n{+}1,1),(n{+}2,1),\dots,(2n,1),(n,1),(n{-}1,1),\dots,(2,1),(n{+}2,2),(n{+}3,2),\dots,(2n,n)),$$

which corresponds to a northeast directed sweep on the matrix $\mathcal{F}H\mathcal{F}$. We refer to the Hamiltonian Jacobi method based on the solution of 4×4 subproblems using sweeps given by the sequence of indices (4.6) as [illegible] 4 [illegible]. By the discussion above, we now expect asymptotic quadratic convergence of the method, and this is what can be observed in experiments; see section 5.

Concerning the computational effort, one can roughly say that one sweep of the improved Hamiltonian 4-Jacobi is approximately twice as expensive as the Hamiltonian 4-Jacobi. On the other hand, one sweep of the Hamiltonian 2-Jacobi needs approximately 80% of the number of flops (floating point operations) of one sweep of the Hamiltonian 4-Jacobi (and thus about 40% of the number of flops of one sweep of the improved Hamiltonian 4-Jacobi).

5. Numerical experiments. We implemented the cyclic nonsymmetric Jacobi algorithm in Matlab Version 7 and ran it on a PC with a Pentium III processor (800 MHz). As a stopping criterion we used $\text{maxoff}(A) := \max_{i>j} |a_{ij}| < 10\texttt{eps}\,\|A\|_2$.

First, the nonsymmetric Jacobi method was run for 100 random complex matrices (generated with the Matlab command `randn`) of size $n \times n$, where $n = 10, 20, \ldots, 150$, after normalization to spectral norm equal to one; see Figure 5.1.

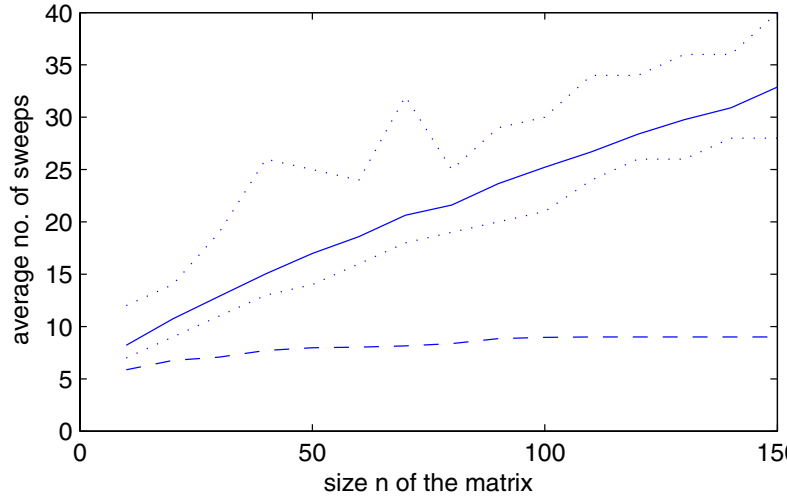


FIG. 5.1. *Test for* 100 *random matrices of different sizes.*

The solid line in Figure 5.1 displays the average number of sweeps needed for convergence for random complex matrices while the dashed line displays the corresponding number for Hermitian matrices. Thus, the method runs much faster for Hermitian matrices than for general complex matrices, an effect that had already been observed by Eberlein [10]. The dotted lines display the maximal and minimal number of sweeps that were needed for convergence for random complex matrices; e.g., for $n = 100$ the algorithm needed between 21 and 30 sweeps. The distribution of the number of sweeps for the tests on 100×100 matrices is shown in Figure 5.2.

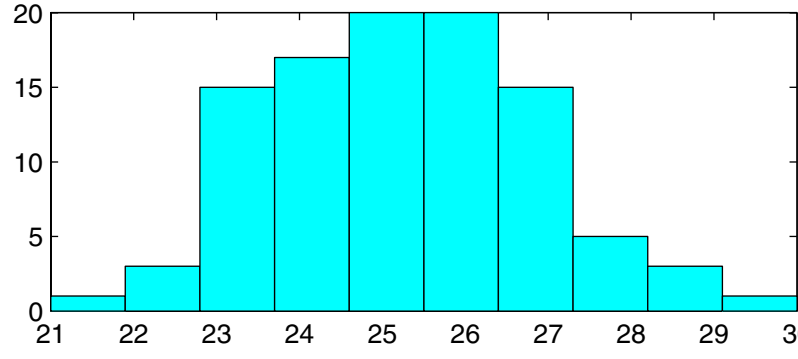


FIG. 5.2. *Distribution of number of sweeps needed for convergence.*

Figure 5.3 displays the typical convergence behavior of the nonsymmetric Jacobi algorithm for a random complex matrix of size 50×50 using top-to-bottom sweeps versus using bottom-to-top sweeps.

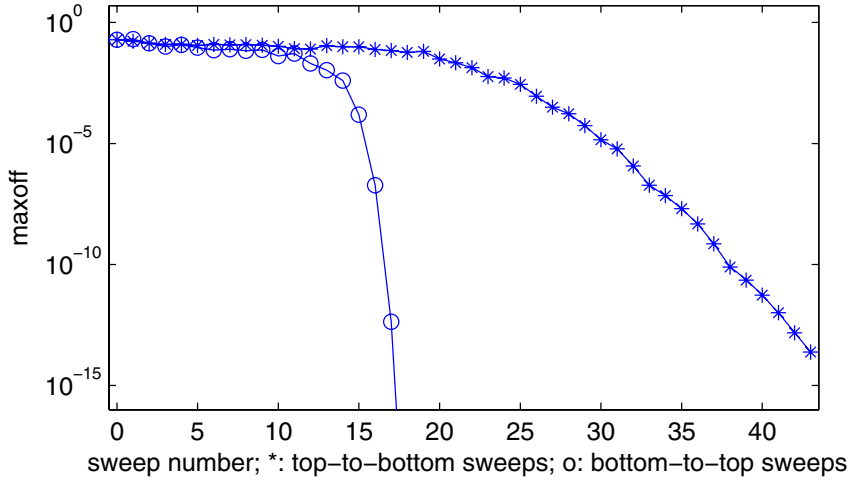


FIG. 5.3. *Typical convergence behavior for general matrices.*

In both cases, the algorithm starts with a phase of "sorting elements" that is characterized by almost stagnation. As expected, it can be observed that maxoff(A) does not decrease monotonically over the number of sweeps. The phase of almost stagnation is followed by the phase of convergence. As predicted by the theory, the convergence rate becomes asymptotically quadratic in the case of bottom-to-top sweeps. However, the convergence rate appears to be only linear in the case of top-to-bottom sweeps.

The situation is completely different when the algorithm is applied to a Hermitian matrix. Figure 5.4 shows the typical convergence behavior of the nonsymmetric Jacobi algorithm for a 50×50 Hermitian matrix. There is hardly any difference in the convergence behavior of the algorithm when using bottom-to-top sweeps compared to using top-to-bottom sweeps and both methods show asymptotic quadratic convergence.

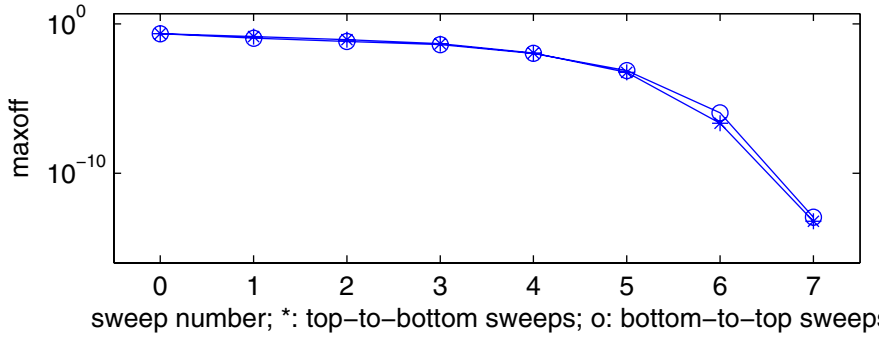


FIG. 5.4. *Typical convergence behavior for Hermitian matrices.*

A second test has been run for 100 matrices close to Schur form. For this, 100 random complex $n \times n$ matrices have been generated and normalized to norm one. Then, the Schur form has been computed by using the Matlab function `schur`, and a random perturbation of norm $1/100$ has been added. Then the nonsymmetric Jacobi algorithm has been run on the perturbed Schur form. The results are displayed in Figure 5.5.

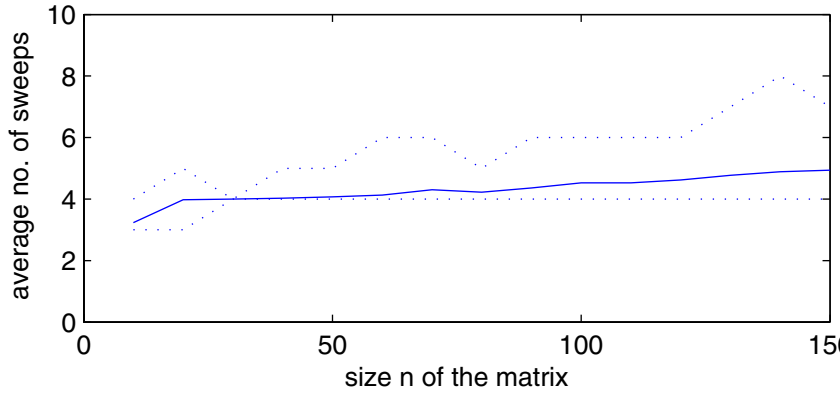


FIG. 5.5. *Test for* 100 *matrices close to Schur form.*

Once again, the solid line displays the average number of sweeps that were needed for convergence, while the dotted lines display the maximal and minimal number of needed sweeps. Even for matrices of size 150×150, the algorithm needs only about five sweeps for convergence, because the entries in the strict lower triangular part of the matrices are of magnitude of order $1/100$ compared to the entries in the upper triangular part, and thus the hypothesis of Theorem 3.3 is very likely to be satisfied so that we may have a quadratic rate of convergence right from the beginning.

Figure 5.6 displays the typical convergence behavior of the nonsymmetric Jacobi algorithm for a matrix of size 50×50 that is close to Schur form. While the algorithm

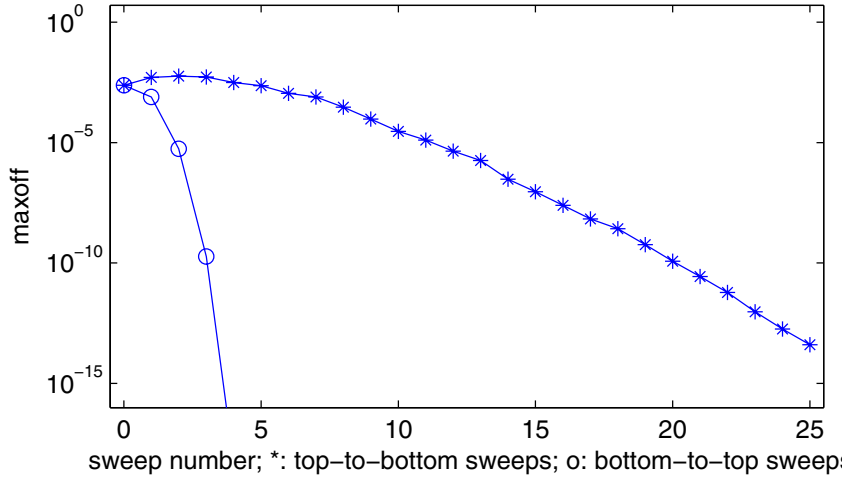


FIG. 5.6. *Typical convergence behavior for matrices close to Schur form.*

almost immediately enters the phase of quadratic convergence when using bottom-to-top sweeps, the convergence rate appears once again to be only linear when using top-to-bottom sweeps. Next, we tested the performance of the Hamiltonian Jacobi algorithms described in section 4 for 100 random Hamiltonian matrices normalized to spectral norm equal to one for the sizes $2n = 10, 20, \ldots, 100$.

Figure 5.7 displays the average number of sweeps that was needed for convergence. The Hamiltonian 2-Jacobi performs similarly to the nonsymmetric Jacobi algorithm for general complex matrices as expected. On the other hand, the Hamiltonian 4-Jacobi needs a much larger number of sweeps. This changes drastically when passing to the improved Hamiltonian 4-Jacobi.

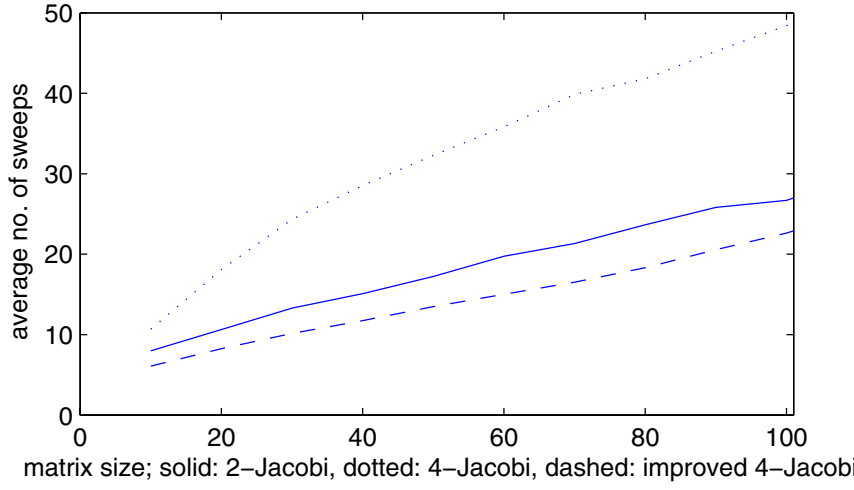


FIG. 5.7. *Test for* 100 *random Hamiltonian matrices of different sizes.*

Figure 5.8 displays the typical convergence behavior of the Hamiltonian Jacobi methods. As expected, the Hamiltonian 2-Jacobi and the improved Hamiltonian 4-

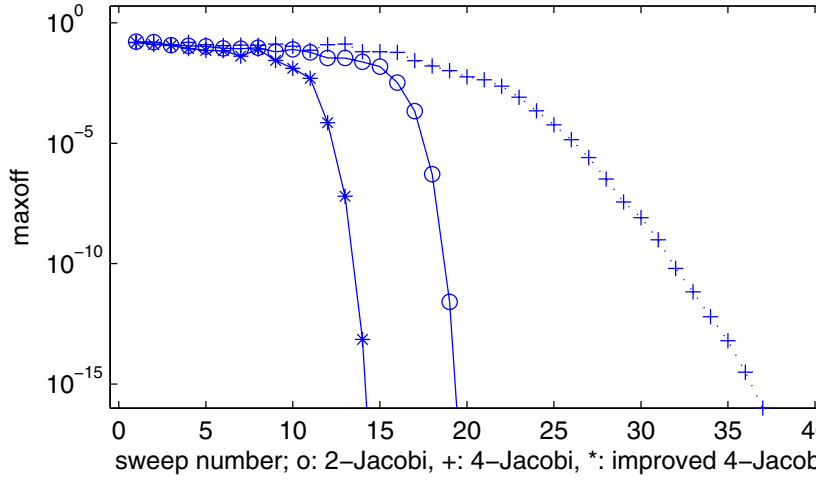


FIG. 5.8. *Typical convergence behavior for Hamiltonian matrices.*

Jacobi show asymptotic quadratic convergence. On the other hand, the asymptotic convergence rate of the Hamiltonian 4-Jacobi appears to be only linear, which explains the large average number of sweeps that is needed for convergence. Although the improved Hamiltonian 4-Jacobi method needs the least number of sweeps for convergence in general, the Hamiltonian 2-Jacobi turns out to be the most efficient version of the Hamiltonian Jacobi methods because the cost of one sweep of the Hamiltonian 2-Jacobi is only about 40% of the cost of one sweep of the improved Hamiltonian 4-Jacobi.

6. Conclusions. We have revisited the nonsymmetric Jacobi algorithm for the computation of the Schur form of a general complex matrix. In particular, asymptotic quadratic convergence of the cyclic version of the method can be proved if northeast directed sweeps are used. Other sweeps, in turn, seem to lead to a linear convergence rate only. Based on this convergence theory, we were able to explain why the convergence behavior of the Hamiltonian Jacobi algorithm proposed in [1] is less satisfactory than expected and what can be done to overcome this inconvenience.

Still, there are many aspects that have not yet been investigated. First, a proof of global convergence is still missing. Then, the convergence behavior of the algorithm is not yet optimal, because the initial phase of almost stagnation is very long and the phase of quadratic convergence is entered rather late. Therefore, preconditioning methods should be introduced and investigated in order to improve the convergence, like it has been done for example, for a Jacobi algorithm for computing the singular value decomposition; see [8]. Another important issue is parallelization. Since the solution of 2×2 subproblems requires only local information, it is possible to implement parallel version of Jacobi-like algorithms, as already considered in [10]. However, the discussion in section 2 shows that further investigation is necessary as a naive parallel implementation of the algorithm may lead to a loss of the property of asymptotic quadratic convergence.

Acknowledgment. I would like to thank Heike Fassbender for valuable discussions and for giving helpful comments on an earlier draft of the paper.

REFERENCES

[1] A. Bunse-Gerstner and H. Fassbender, *A Jacobi-like method for solving algebraic Riccati equations on parallel computers*, IEEE Trans. Automat. Control, 42 (1997), pp. 1071–1084.
[2] R. Byers, *A Hamiltonian QR-algorithm*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 212–229.
[3] R. Byers, *A Hamiltonian Jacobi algorithm*, IEEE Trans. Automat. Control, 35 (1990), pp. 566–570.
[4] R. Causey, *Computing eigenvalues of non-Hermitian matrices by methods of Jacobi type*, J. SIAM, 6 (1958), pp. 172–181.
[5] J. Charlier and P. van Dooren, *A Jacobi-like algorithm for computing the generalized Schur form of a regular pencil*, J. Comput. Appl. Math., 27 (1989), pp. 17–36.
[6] D. Chu, X. Liu, and V. Mehrmann, *A numerical method for computing the Hamiltonian Schur form*, Numer. Math., 105 (2007), pp. 375–412.
[7] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.
[8] Z. Drmač, *A posteriori computation of the singular vectors in a preconditioned Jacobi SVD algorithm*, IMA J. Numer. Anal., 19 (1999), pp. 191–213.
[9] P. Eberlein, *A Jacobi-like method for the automatic computation of eigenvalues and eigenvectors of an arbitrary matrix*, J. SIAM, 10 (1962), pp. 74–88.
[10] P. Eberlein, *On the Schur decomposition of a matrix for parallel computation*, IEEE Trans. Comp., 36 (1987), pp. 167–174.

[11] H. Fassbender, D. S. Mackey, and N. Mackey, *Hamiltonian and Jacobi come full circle: Jacobi algorithms for structured Hamiltonian eigenproblems*, Linear Algebra Appl., 332–334 (2001), pp. 37–80.
[12] H. Goldstine and L. Horwitz, *A procedure for the diagonalization of normal matrices*, J. ACM, 6 (1959), pp. 176–195.
[13] J. Greenstadt, *A method for finding roots of arbitrary matrices*, Math. Tables Aids Comput., 9 (1955), pp. 47–52.
[14] J. Greenstadt, *Some numerical experiments in triangularizing matrices*, Numer. Math., 4 (1962), pp. 187–195.
[15] D. Hacon, *Jacobi's method for skew-symmetric matrices*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 619–628.
[16] V. Hari, *A Jacobi-like eigenvalue algorithm for general real matrices*, Glas. Mat. Ser. III, 11 (1976), pp. 367–378.
[17] P. Henrici, *On the speed of convergence of cyclic and quasicyclic Jacobi methods for computing the eigenvalues of hermitian matrices*, J. SIAM, 6 (1958), pp. 144–162.
[18] A. Hilliges, C. Mehl, and V. Mehrmann, *On the solution of palindromic eigenvalue problems*, in Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS), Jyväskylä, Finland, 2004.
[19] C. P. Huang, *A Jacobi-type method for triangularizing an arbitrary matrix*, SIAM J. Numer. Anal., 4 (1975), pp. 566–570.
[20] K. Hüper, *A Calculus Approach to Matrix Eigenvalue Algorithms*, Habilitationsschrift, Universität Würzburg, Würzburg, Germany, 2002.
[21] C. G. J. Jacobi, *Über ein leichtes Verfahren, die in der Theorie der Säcularströmungen vorkommenden Gleichungen numerisch aufzulösen*, J. Reine Angew. Math., 30 (1846), pp. 51–95.
[22] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Clarendon Press, Oxford, UK, 1995.
[23] W.-W. Lin, V. Mehrmann, and H. Xu, *Canonical forms for Hamiltonian and symplectic matrices and pencils*, Linear Algebra Appl., 302/303 (1999), pp. 469–533.
[24] M. Lotkin, *Characteristic values of arbitrary matrices*, Quart. Appl. Math., 14 (1956), pp. 267–275.
[25] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Numerical methods for palindromic eigenvalue problems: Computing the anti-triangular Schur form*, Technical report 409, DFG Research Center Matheon, Berlin, 2007.
[26] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Structured polynomial eigenvalue problems: Good vibrations from good linearizations*, SIAM Matrix Anal. Appl., 28 (2006), pp. 1029–1051.
[27] C. Mehl, *Jacobi-like algorithms for the indefinite generalized Hermitian eigenvalue problem*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 964–985.
[28] V. Mehrmann, *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*, Lecture Notes in Control and Inform. Sci. 163, Springer-Verlag, Heidelberg, July 1991.
[29] M. H. C. Paardekooper, *An eigenvalue algorithm for skew-symmetric matrices*, Numer. Math., 17 (1971), pp. 189–202.
[30] C. Paige and C. Van Loan, *A Schur decomposition for Hamiltonian matrices*, Linear Algebra Appl., 14 (1981), pp. 11–32.
[31] A. Schönhage, *Zur quadratischen Konvergenz des Jacobi-Verfahrens*, Numer. Math., 6 (1964), pp. 410–412.
[32] G. W. Stewart, *A Jacobi-like algorithm for computing the Schur decomposition of a non-Hermitian matrix*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 853–864.
[33] G. W. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
[34] K. Veselić, *A Jacobi eigenreduction algorithm for definite matrix pairs*, Numer. Math., 64 (1993), pp. 241–269.
[35] J. H. Wilkinson, *Note on the quadratic convergence of the cyclic Jacobi process*, Numer. Math, 4 (1962), pp. 296–300.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 312–323

A CLASS OF SPARSE UNIMODULAR MATRICES GENERATING MULTIRESOLUTION AND SAMPLING ANALYSIS FOR DATA OF ANY LENGTH*

N. D. ATREAS†, C. KARANIKAS†, AND P. POLYCHRONIDOU†

Abstract. We introduce a class of sparse unimodular matrices U^m of order $m \times m$, $m = 2, 3, \ldots$. Each matrix U^m has all entries 0 except for a small number of entries 1. The construction of U^m is achieved by iteration, determined by the prime factorization of a positive integer m and by new dilation operators and block matrix operators. The iteration above gives rise to a multiresolution analysis of the space V_m of all m-periodic complex-valued sequences, suitable to reveal information at different scales and providing sampling formulas on the multiresolution subspaces of V_m. We prove that the matrices U^m are invertible, and we present a recursion equation to compute the inverse matrices. Finally, we connect the transform induced by the matrix U^m with the underlying natural tree structure and random walks on trees.

Key words. sparse matrices, multiresolution analysis, sampling theory, tree structures

AMS subject classifications. 65F10, 65F30, 15A09, 68P20, 94A20

DOI. 10.1137/050638679

1. Introduction.

The evolution of large digital libraries has dramatically transformed the processing, storage, and retrieval of information (see [2]). In order to handle a large amount of information, we need fast computations and storage saving, and the role of sparse matrices to both aforementioned requirements becomes very important. Since sparse matrices have a "small" number of nonzero elements, a variety of techniques have been developed for storing and processing them efficiently (see [4], [8]). Sparse matrices are used mainly in combinatorics and in application areas such as graph theory, for describing a low density of significant data or connections, and for numerical solution of linear partial differential equations, where the partial differentiable operators are approximated by finite difference operators, giving rise to a system of equations which involves a sparse coefficient matrix. It is natural to wonder whether we can build sparse matrices for extracting local information.

One of the widely used tools for extracting local information is multiresolution analysis (MRA). A multiresolution analysis of a Hilbert space V is a nested sequence of closed, shift invariant subspaces $\{V_i \subset V_{i+1} : i = 0, 1, 2 \ldots\}$ of V whose union is dense in V such that the subspaces V_i are scaled versions of each other under the action of a dilation operator $D : V_i \to V_{i+1}$. As a consequence, each subspace V_i reveals information at different scales/resolutions, and the resolution of the subspace V_{i+1} is higher than the resolution of the subspace V_i (see [5] for an overview of MRAs on various spaces). We notice here that most MRAs provide sampling formulas associated to the subspaces V_i (see [9], [10]). In [1], we presented the sampling formula associated to the subspaces V_i of a MRA of the space of p^M-periodic sequences. Recall that an M-dimensional subspace W of a space V of sequences of length N has a [illegible] $\{s_0, \ldots, s_{M-1}\}$, if there exist M positive integers $0 \le n_1 < \cdots < n_M < N$ such

*Received by the editors August 23, 2005; accepted for publication (in revised form) by D. Calvetti October 15, 2007; published electronically March 19, 2008. This research was supported by the GSRT program "Pythagoras II."

http://www.siam.org/journals/simax/30-1/63867.html

†Department of Informatics, Aristotle University of Thessaloniki, 54-124, Thessaloniki, Greece (natreas@csd.auth.gr, karanika@csd.auth.gr, ppolychr@csd.auth.gr).

that for any sequence $\mathbf{t} = \{t_1, \ldots, t_N\} \in W$ we have

$$t(n) = \sum_{j=0}^{M-1} t(n_j) s_j(n),\ 0 \le n \le N-1.$$

In particular, we say that W has a s, if there exist M positive integers $0 \le n_1 < \cdots < n_M < N$ such that for any sequence $\mathbf{t} \in W$ we have

$$t(n) = \sum_{j=0}^{M-1} t(n_j) s(n - n_j),\ 0 \le n \le N-1.$$

In both cases, we say that W is a of V.

Another tool used for extracting local information is a tree transform. Local information in a tree is stored in , starting with a root node and ending with terminal nodes called . For example, a binary tree transform associated to data $T = \{t_1, \ldots, t_{2^N}\}$ is a collection of numbers:

$$\boldsymbol{\alpha} = \{\alpha_{n,k}(T),\ k = 1, \ldots, 2^n,\ n = 0, \ldots, N\}.$$

The collection $\boldsymbol{\alpha}$ has a binary tree structure with $N+1$ resolution levels. Each resolution level cuts T into 2^n equal pieces. If we denote

$$T(n,k) = \{t_s,\ s = (k-1)2^{N-n} + 1, \ldots, k2^{N-n}\},$$

then the number $a_{0,1}(T)$ corresponds to the initial node of the tree; the number $a_{n,k}(T)$ corresponds to the k-node of the nth generation and encodes the information associated to the subset $T(n,k)$. In [7], one of the authors has developed a discrete tree transform (DTT), which has been used for a pattern recognition process (see [3]).

In order to build our sparse matrices U^m of order $m \times m$, we exploit the basic properties of a MRA construction on matrices.

In section 2, a construction is presented for U^m in different scales by an iteration process, determined by the prime factorization of a positive integer m : $m = p_1 p_2 \ldots p_N$ $(p1 \ge p2 \ldots \ge p_N)$ and by repetitive dilation and block-matrix operations (see Definitions 1–6). Our construction starts with the matrix $U^m(0) = \{1\}$. $U^m(n+1)$ is a block matrix obtained from joining two matrices: a matrix derived by a dilation process on $U^m(n)$ and a properly selected permutation matrix. In the Nth scale the resulting matrix $U^m(N) = U^m$ is a $(0,1)$-matrix admitting the following properties:

- It is a unimodular matrix (see Proposition 1);
- the inverse matrix $(U^m)^{-1}$ is also a sparse matrix with entries $1, 0, -1$ constructed by a recursion equation on matrices, which can be easily implemented via an algorithm (see Theorem 1).

In section 3, we see that the matrices U^m encode local information. In fact:

- in Definition 7 we see that the matrix U^m gives rise to a multiresolution analysis $\{W_0 \subset \ldots \subset W_N = V_m\}$ of the space V_m of all m-periodic complex-valued sequences. The subspaces W_i are spanned by properly selected row vectors of U_m and display information at different resolution levels. In addition, in Theorem 2 we prove that the resolution subspaces W_i are sampling subspaces, and we present the sampling formula for W_i.

- The transform $L_m : V_m \to V_m,\ L_m(\mathbf{t}) = U^m\mathbf{t}$ has tree structure. In fact, in (8) we established its connection with the discrete tree transform given in [7].

2. Construction and properties of U^m. (see also [6]): Let $\mathsf{M}_{n,m}$ be the set of all $n \times m$ matrices over the field of complex numbers. If $n = m$, then $\mathsf{M}_{n,m}$ is abbreviated to M_n. We shall use the symbolism $A = [A_{ij}]$ to denote a matrix A with elements A_{ij}. The notation

$$A_i = \{A_{ij} : j = 1, \dots, m\}$$

shall be used to denote the i-row of a matrix $A \in \mathsf{M}_{n,m}$, and often it will be referred to as the ith of the matrix A. We use the notation A^T to denote the transpose of a matrix A. A square matrix $A \in \mathsf{M}_n$ is invertible, if there is a unique square matrix $A^{-1} \in \mathsf{M}_n$ called the inverse matrix of A such that $AA^{-1} = \mathbf{I}_n$, where $\mathbf{I}_n$ is the identity matrix. A matrix having a small number of nonzero elements is called $P \in \mathsf{M}_n$ is a matrix, if it is formed from the identity matrix $\mathbf{I}_n$ by reordering its columns (or rows). The determinant of a permutation matrix P is given by:

$$\mathrm{Det}(P) = \mathrm{sgn}\,\sigma,$$

where $\sigma = \{\sigma(i) : i = 1, \dots, n\}$ is the permutation of its columns and the signature $\mathrm{sgn}\,\sigma$ equals $(-1)^r$, where r is the number of transpositions of pairs of columns that must be composed to build up the permutation. In practice, in order to estimate r we compute the number of elements $\sigma(i) : \sigma(1) > \sigma(i), i = 2, \dots, n$, then we compute the number of elements $\sigma(i) : \sigma(2) > \sigma(i),\ i = 3, \dots, n$, etc., and finally we sum all previously computed numbers. A square matrix with determinant ± 1 is called

The of a real number x shall be denoted by $\lceil x \rceil = \inf\{n \in \mathbf{Z} : x \le n\}$ ($\mathbf{Z}$ is the set of integers). If p, q are natural numbers, we denote by $\mathrm{Mod}(p, q)$ the remainder on division of p by q, and we shall use the symbolism $[q]_p = \{q + tp : t \in \mathbf{Z}\}$ to denote the residue class of **q modulo p**.

We consider the unique (up to a rearrangement of factors) prime factorization of a positive integer m:

$$m = p_1 p_2 \dots p_N, \tag{1}$$

where $p_1 \ge p_2 \ge \dots \ge p_N$, and we denote:

$$J(0) = 1,\ \ J(n) = \prod_{i=1}^{n} p_i,\ \ n = 1, \dots, N, \tag{1a}$$

$$A(i) = \prod_{r=i}^{N} p_r,\ \ i = 1, \dots, N,\ \ A(N+1) = 1. \tag{1b}$$

We define the following dilation operators D_p and H_p on the set $\mathsf{M}_{n,m}$, where $p = 2, 3, \dots$.

Definition 1. $D_p : \mathsf{M}_{n,m} \to \mathsf{M}_{n,pm}$

$$D_p(M) = \left\{ M_{i,\lceil \frac{j}{p} \rceil},\ \ i = 1, \dots, n,\ \ j = 1, \dots, pm \right\}.$$

Notice that D_p can be written as a block matrix:

$$D_p(M) = \begin{pmatrix} D_p(M_{11}) & \dots & D_p(M_{1m}) \\ \vdots & \ddots & \vdots \\ D_p(M_{n1}) & \dots & D_p(M_{nm}) \end{pmatrix}, \tag{2}$$

where $D_p(M_{ij}) \in \mathsf{M}_{1,p}$: $D_p(M_{ij}) = \{M_{ij}, M_{ij}, \dots, M_{ij}\}$.

$$D_2\left(\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\right) = \begin{pmatrix} a_{11} & a_{11} & a_{12} & a_{12} \\ a_{21} & a_{21} & a_{22} & a_{22} \end{pmatrix},$$

$$D_3\left(\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\right) = \begin{pmatrix} a_{11} & a_{11} & a_{11} & a_{12} & a_{12} & a_{12} \\ a_{21} & a_{21} & a_{21} & a_{22} & a_{22} & a_{22} \end{pmatrix}.$$

Definition 2. $H_p : \mathsf{M}_{n,m} \to \mathsf{M}_{pn,m}$

$$H_p(M) = \left\{ \begin{array}{ll} M_{\lceil \frac{i}{p} \rceil, j}, & \text{whenever } i \in [0]_p, \\ 0, & \text{whenever } i \notin [0]_p, \end{array} \quad i = 1, \dots, pn, j = 1, \dots, m \right\}.$$

$$H_2\left(\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\right) = \begin{pmatrix} 0 & 0 \\ a_{11} & a_{12} \\ 0 & 0 \\ a_{21} & a_{22} \end{pmatrix}, \quad H_3\left(\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\right) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ a_{11} & a_{12} \\ 0 & 0 \\ 0 & 0 \\ a_{21} & a_{22} \end{pmatrix}.$$

Definition 3. $S(.,\dots,.) : \mathsf{M}_{n_1,m} \times \cdots \times \mathsf{M}_{n_k,m} \to \mathsf{M}_{n_1+\cdots+n_k,m}$

$$S(M_1, \dots, M_k) = \begin{pmatrix} M_1 \\ \\ M_k \end{pmatrix}.$$

Definition 4. $Q_p : \mathsf{M}_{n,m} \to \mathsf{M}_{pn,pm}$ $(p \geq 1)$

$$Q_p(M) = \underbrace{M \oplus \cdots \oplus M}_{p\text{-times}} = \begin{pmatrix} M & \boldsymbol{O} \\ \\ \boldsymbol{O} & M \end{pmatrix}, \tag{3}$$

$\oplus$ direct sum $\boldsymbol{O}$ $n \times m$

Definition 5. p_1, p_2 $p_2 > 1$ $R(p_1, p_2) \in \mathsf{M}_{p_1(p_2-1),p_1p_2}$

$$R(p_1, p_2) = S\left(Q_1\left(\mathbf{e}_1^{p_2}\right), \dots, Q_{p_1}\left(\mathbf{e}_{p_2-1}^{p_2}\right)\right),$$

$\boldsymbol{e}_i^{p_2}$ i $\boldsymbol{I}_{p_2}$

Let $p_1 = 2$, $p_2 = 2$; then $\mathbf{e}_1^2 = \{1, 0\}$, $\mathbf{O} = \{0, 0\}$ and

$$R(2,2) = Q_2\left(\mathbf{e}_1^2\right) = \begin{pmatrix} \mathbf{e}_1^2 & \mathbf{O} \\ \mathbf{O} & \mathbf{e}_1^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Let $p_1 = 2$, $p_2 = 3$; then $\mathbf{e}_1^3 = \{1, 0, 0\}$, $\mathbf{e}_2^3 = \{0, 1, 0\}$, $\mathbf{O} = \{0, 0, 0\}$, and

$$R(2,3) = S\left(Q_2\left(\mathbf{e}_1^3\right), Q_2\left(\mathbf{e}_2^3\right)\right) = \begin{pmatrix} \mathbf{e}_1^3 & \mathbf{O} \\ \mathbf{O} & \mathbf{e}_1^3 \\ \mathbf{e}_2^3 & \mathbf{O} \\ \mathbf{O} & \mathbf{e}_2^3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Remark 1. (i) Let r, s be positive integers. It is easy to see that $D_r D_s(M) = D_{rs}(M)$. The same is also true for the operators H_p and Q_p.

(ii) Since the matrix $Q_{p_1}\left(\mathbf{e}_{p_2}^{p_2}\right)$ has not been used for the construction of the matrix $R(p_1, p_2)$, we have $(R(p_1, p_2))_{\cdot, lp_2} = 0$, for any $l = 1, \ldots, p_1$.

Definition 6. [illegible] (1) [illegible] m, [illegible] $U^m(n) \in M_{J(n)}$ ($J(n)$ [illegible] (1a)), [illegible] $n = 0, \ldots, N$, [illegible]

$$U^m(n) = \begin{cases} \{1\}, & n = 0, \\ S\left(D_{p_1}\left(U^m(0)\right), R(1, p_1)\right), & n = 1, \\ S\left(D_{p_n}\left(U^m(n-1)\right), R\left(J(n-1), p_n\right)\right), & n = 2, \ldots, N. \end{cases}$$

[illegible] $n = N$, [illegible] $U^m(N) = U^m$ [illegible]

$$U^p(1) = \begin{pmatrix} D_p(\{1\}) \\ R(1,p) \end{pmatrix} = \begin{pmatrix} 1 & 1 & \ldots & 1 & 1 \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix}. \tag{4}$$

Let $m = 12 = p_1 p_2 p_3$, where $p_1 = 3$, $p_2 = 2$, $p_3 = 2$; then $N = 3$, and we have

$$U^{12}(1) = \begin{pmatrix} 1&1&1 \\ 1&0&0 \\ 0&1&0 \end{pmatrix}, \quad U^{12}(2) = \begin{pmatrix} 1&1&1&1&1&1 \\ 1&1&0&0&0&0 \\ 0&0&1&1&0&0 \\ 1&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&0&1&0 \end{pmatrix}, \quad U^{12}(3) = \begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&1&0&0&0&0 \\ 1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \end{pmatrix}.$$

Now let $j = 1, \ldots, p_n - 1$, and we define the following column matrices $V_j^{p_n} = \{v_{kj}^{p_n} : k = 1, \ldots, p_n\}$:

$$v_{kj}^{p_n} = \begin{cases} 1 & \text{whenever } k = j, \\ -1 & \text{whenever } k = p_n, \\ 0 & \text{elsewhere.} \end{cases} \tag{5}$$

PROPOSITION 1. [illegible] $\{p_n : n = 1, \dots, N\}$ [illegible] m [illegible] (1) [illegible] $J(n)$ [illegible] (1a) [illegible]

$$\operatorname{Det}\left(U^m(n)\right) = \begin{cases} (-1)^{p_1+1}, & n = 1, \\ (-1)^{q_n} \operatorname{Det}(U^m(n-1)), & n = 2, \dots, N, \end{cases}$$

[illegible] $q_n = \frac{p_n - 1}{4} J(n-1)\left(J(n) - p_n + 4\right)$ [illegible] $U^m(n)$ unimodular [illegible]. For $n = 1$, we use (4) to get $\operatorname{Det}(U^m(1)) = (-1)^{1+p_1} \operatorname{Det}(M^{1,p_1})$, where M^{1,p_1} is a minor of the matrix $U^m(1)$. Since $M^{1,p_1} = \mathbf{I}_{p_1-1}$, we get $\operatorname{Det}(U^m(1)) = (-1)^{1+p_1}$.

Let $n > 1$, and let $\mathbf{e}_i^{p_n}$ be the ith row vector of the identity matrix $\mathbf{I}_{p_n}$; we consider the following block matrix $C(n) \in \mathsf{M}_{J(n)}$:

$$C(n) = \begin{pmatrix} Q_{J(n-1)}\left(\left(\mathbf{e}_{p_n}^{p_n}\right)^T\right) & Q_{J(n-1)}\left(V_1^{p_n}\right) & \dots & Q_{J(n-1)}\left(V_{p_n-1}^{p_n}\right) \end{pmatrix},$$

where the block submatrices $Q_{J(n-1)}((\mathbf{e}_{p_n}^{p_n})^T)$ and $Q_{J(n-1)}\left(V_j^{p_n}\right)$, $j = 1, \dots, p_n - 1$, are in $\mathsf{M}_{J(n),J(n-1)}$ (the column matrices $V_j^{p_n}$ are given in (5)). The block-matrix multiplication $U^m(n)C(n)$ derives the following block diagonal matrix (for a proof of (6), see Appendix A):

$$\begin{aligned} U^m(n)C(n) &= \begin{pmatrix} D_{p_n}\left(U^m(n-1)\right) \\ Q_{J(n-1)}\left(\mathbf{e}_1^{p_n}\right) \\ \vdots \\ Q_{J(n-1)}\left(\mathbf{e}_{p_n-1}^{p_n}\right) \end{pmatrix} \\ &\qquad \begin{pmatrix} Q_{J(n-1)}\left(\left(\mathbf{e}_{p_n}^{p_n}\right)^T\right) & Q_{J(n-1)}\left(V_1^{p_n}\right) & \dots & Q_{J(n-1)}\left(V_{p_n-1}^{p_n}\right) \end{pmatrix} \\ &= \begin{pmatrix} U^m(n-1) & & & \mathbf{O} \\ & \mathbf{I}_{J(n-1)} & & \\ & & \ddots & \\ \mathbf{O} & & & \mathbf{I}_{J(n-1)} \end{pmatrix}, \end{aligned} \tag{6}$$

where the zero matrix $\mathbf{O}$ in the right-hand side of (6) belongs in $\mathsf{M}_{J(n-1)}$. As a result we get

$$\operatorname{Det}(U^m(n)) \operatorname{Det}(C(n)) = \operatorname{Det}(U^m(n-1)).$$

The computation of $\operatorname{Det}(C(n))$ is equivalent to computing $\operatorname{Det}(K(n))$, where

$$K(n) = \begin{pmatrix} Q_{J(n-1)}\left(\left(\mathbf{e}_{p_n}^{p_n}\right)^T\right) & Q_{J(n-1)}\left(\left(\mathbf{e}_1^{p_n}\right)^T\right) & \dots & Q_{J(n-1)}\left(\left(\mathbf{e}_{p_n-1}^{p_n}\right)^T\right) \end{pmatrix}$$

is a block matrix in $\mathsf{M}_{J(n)}$ resulting from $C(n)$ by replacing each block submatrix

$Q_{J(n-1)}\left(V_j^{p_n}\right)$ with the linear combination:

$$Q_{J(n-1)}\left(V_j^{p_n}\right) + Q_{J(n-1)}\left(\left(\mathbf{e}_{p_n}^{p_n}\right)^T\right) = Q_{J(n-1)}\left(\left(\mathbf{e}_j^{p_n}\right)^T\right), \quad j = 1, \ldots, p_n - 1.$$

$K(n)$ is a permutation matrix, so $\mathrm{Det}(K(n)) = \mathrm{sgn}\ \sigma_n$, where σ_n is the permutation of its columns, and thus

$$\mathrm{Det}(U^m(n)) = \mathrm{sgn}\ \sigma_n\ \mathrm{Det}(U^m(n-1)).$$

The permutation $\sigma_n = \{\sigma_n(1), \ldots, \sigma_n(J(n))\}$ of the columns of the matrix $K(n)$ can be written as:

$$\sigma_n = \Upsilon_{0,n} \bigcup_{i=1}^{p_n - 1} \Upsilon_{i,n},$$

where $\Upsilon_{0,n} = \{tp_n : 1 \le t \le J(n-1)\}$ and $\Upsilon_{i,n} = \{i + tp_n : 0 \le t \le J(n-1) - 1\}$, $i \ge 1$. In Appendix B we prove that $\mathrm{sgn}\ \sigma_n = (-1)^{q_n}$, where $q_n = \frac{p_n - 1}{4} J(n-1)\,(J(n) - p_n + 4)$, and we complete the proof. □

THEOREM 1. [illegible] $U^m(n)$ [illegible]

$$(U^m(n))^{-1} = \begin{cases} \{1\}, & n = 0, \\ \left(H_{p_n}\left((U^m(n-1))^{-1}\right) \quad Q_{J(n-1)}(V_1^{p_n}) \quad \ldots \quad Q_{J(n-1)}(V_{p_n-1}^{p_n})\right), & n = 1, \ldots, N \end{cases}$$

[illegible] We multiply both sides of (6) with the block diagonal matrix

$$\begin{pmatrix} (U^m(n-1))^{-1} & & & \mathbf{O} \\ & \mathbf{I}_{J(n-1)} & & \\ & & \ddots & \\ \mathbf{O} & & & \mathbf{I}_{J(n-1)} \end{pmatrix},$$

whose block submatrices are in $\mathsf{M}_{J(n-1)}$, and we deduce that the inverse matrix $(U^m(n))^{-1}$ results from the following block-matrix multiplication:

$$\begin{aligned} (U^m(n))^{-1} &= \left(Q_{J(n-1)}\left(\left(\mathbf{e}_{p_n}^{p_n}\right)^T\right) \quad Q_{J(n-1)}\left(V_1^{p_n}\right) \quad \ldots \quad Q_{J(n-1)}\left(V_{p_n-1}^{p_n}\right)\right) \\ &\quad \cdot \begin{pmatrix} (U^m(n-1))^{-1} & & & \mathbf{O} \\ & \mathbf{I}_{J(n-1)} & & \\ & & \ddots & \\ \mathbf{O} & & & \mathbf{I}_{J(n-1)} \end{pmatrix} \\ &= \left(Q_{J(n-1)}\left(\left(\mathbf{e}_{p_n}^{p_n}\right)^T\right) \cdot (U^m(n-1))^{-1} \quad Q_{J(n-1)}\left(V_1^{p_n}\right) \quad \ldots \quad Q_{J(n-1)}\left(V_{p_n-1}^{p_n}\right)\right) \\ &= \left(H_{p_n}\left(\mathbf{I}_{J(n-1)}\right) \cdot (U^m(n-1))^{-1} \quad Q_{J(n-1)}\left(V_1^{p_n}\right) \quad \ldots \quad Q_{J(n-1)}\left(V_{p_n-1}^{p_n}\right)\right) \\ &= \left(H_{p_n}\left((U^m(n-1))^{-1}\right) \quad Q_{J(n-1)}\left(V_1^{p_n}\right) \quad \ldots \quad Q_{J(n-1)}\left(V_{p_n-1}^{p_n}\right)\right). \quad \square \end{aligned}$$

$$\left(U^{12}(2)\right)^{-1} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & -1 & 0 & 0 & -1 \end{pmatrix},$$

$$\left(U^{12}(3)\right)^{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}.$$

3. The matrices U^m and multiresolution-type sampling spaces. Let V_m be the space consisting of all m-periodic complex-valued sequences, where m satisfies the prime factorization (1), i.e., $m = p_1 p_2 \dots p_N$, where $p_1 \geq p_2 \geq \cdots \geq p_N$, and then there exists a unique sequence of scalars $\boldsymbol{\alpha} = \{\alpha_n : n = 1, \dots, m\}$ such that any element $\mathbf{t} = \{t(n) : n = 1, \dots, m\} \in V_m$ can be written as:

$$\mathbf{t} = \sum_{i=1}^{m} \alpha_i U_i^m,$$

where U_i^m is the ith row vector of the matrix U^m. In addition, there holds $\alpha_i = \langle t(.), ((U^m)^{-1})_{.,i} \rangle$, where $\langle ., . \rangle$ is the usual inner product of V_m.

DEFINITION 7. $\{J(n) : n = 1, \dots, N\}$ (1a) $\{W_0 \subset \cdots \subset W_N = V_m\}$ V_m $\{U_r^m : r = 1, \dots, J(i)\}$ W_i multiresolution analysis V_m

Let

$$B(i) = \begin{pmatrix} U_1^m \\ \vdots \\ U_{J(i)}^m \end{pmatrix}, \; i = 0, \dots, N, \tag{7}$$

then $B(i) \in \mathsf{M}_{J(i),m}$, and the following applies.

LEMMA 1. $B(i) = S\left(D_m(\{1\}), \left\{D_{A(k+2)}(R(J(k), p_{k+1})), k = 0, \dots, i-1\right\}\right)$, $i \geq 1$ $\{A(i) : i = 1, \dots, N\}$ (1b)

. From the recursion equation of Definition 6, we have

$$\text{for } n = 1: \;\; U^m(1) = S\left(D_{p_1}(\{1\}), R(1, p_1)\right),$$

$$\text{for } n = 2: \;\; U^m(2) = S\left(D_{p_1 p_2}(\{1\}), D_{p_2}(R(1, p_1)), R(J(1), p_2)\right), \dots,$$

$$\text{for } n = N: \;\; U^m = S\left(D_m(\{1\}), D_{p_2 \dots p_N}(R(1, p_1)), D_{p_3 \dots p_N}(R(J(1), p_2)), \; \dots,\right.$$

$$\left. D_{p_N}(R(J(N-2), p_{N-1})), R(J(N-1), p_N)\right)$$

$$= S\left(D_m(\{1\}), D_{A(2)}(R(J(0), p_1)), \dots, D_{A(N+1)}(R(J(N-1), p_N))\right).$$

We observe that $D_m(\{1\}) \in \mathsf{M}_{1,m}$, $D_{A(2)}(R(J(0),p_1)) \in \mathsf{M}_{p_1-1,m}, \ldots, D_{A(i+1)}(R(J(i-1),p_i)) \in \mathsf{M}_{J(i)-J(i-1),m}$, and we have the result. □

THEOREM 2. [illegible] W_i $(i = 0, \ldots, N)$ [illegible] 7 [illegible] $\boldsymbol{t} = \{t(n) : n = 1, \ldots, m\} \in W_i$ [illegible]

$$t(n) = \sum_{k=1}^{J(i)} t(kA(i+1))\, c(n-(k-1)A(i+1)),$$

[illegible] $c(n) = \begin{cases} 1, & n = 1, \ldots, A(i+1) \\ 0, & \text{elsewhere} \end{cases}$ [illegible] $\{A(k) : k = 1, \ldots, N+1\}$ [illegible] (1b)

[illegible]. Since the space W_i is the span of the first $J(i)$ rows of the matrix U^m, there exists a sequence of scalars $\mathbf{d} = \{d_1, \ldots, d_{J(i)}\}$:

$$\mathbf{t} = \sum_{k=1}^{J(i)} d_k U_k^m = \mathbf{d}B(i),$$

where the matrix $B(i)$ is defined in (7). From Lemma 1 and (2) we deduce that

$$B(i) = D_{A(i+1)}(M_i),$$

where

$$M_i = S\left(D_{J(i)}(\{1\}), \left\{D_{p_{k+2}\ldots p_i}(R(J(k),p_{k+1})), k = 0, \ldots, i-2\right\}, R(J(i-1),p_i)\right),\ i \geq 1,$$

$M_i \in \mathsf{M}_{J(i)}$, and we use Definition 1 to get

$$\mathbf{t} = \mathbf{d}B(i) = \mathbf{d}D_{A(i+1)}(M_i) = \left\{\mathbf{d}(M_i)_{\cdot,\left\lceil \frac{n}{A(i+1)} \right\rceil} : n = 1, \ldots, m\right\}.$$

We define $\alpha_j = \mathbf{d}(M_i)_{\cdot,j}$ $(j = 1, \ldots, J(i))$, so we have

$$t(n) = \left\{a_{\left\lceil \frac{n}{A(i+1)} \right\rceil} :\ n = 1, \ldots, m\right\}.$$

Let $c(n) = \begin{cases} 1, & n = 1, \ldots, A(i+1) \\ 0, & \text{elsewhere} \end{cases}$ be an m-periodic sequence; then we can write:

$$t(n) = \sum_{k=1}^{J(i)} \alpha_k c(n-(k-1)A(i+1)),$$

and we observe that whenever $n = sA(i+1),\ s = 1, \ldots, J(i)$, then

$$t(sA(i+1)) = \sum_{k=1}^{J(i)} \alpha_k c((s-(k-1))A(i+1)) = \sum_{k=1}^{J(i)} \alpha_k \delta_{s-(k-1),1} = \alpha_s,$$

so the theorem is proved. □

[illegible] Let $\mathbf{t} = \{t_1, \ldots, t_{12}\}$. Since $12 = 3\ 2\ 2$, we have $p_1 = 3$, $p_2 = 2$, and $p_3 = 2$, and the matrix U^{12} is presented above. Obviously, $A(1) = 12$, $A(2) = 4$,

$A(3) = 2$, so if $Pi(\mathbf{t})$ is the orthogonal projection of $\mathbf{t}$ into the subspaces $W_i,\ i = 0, 1, 2$, then by Theorem 2 we have

$$P_0(\mathbf{t}) = \{t_{12}, t_{12} \ldots, t_{12}\},$$

$$P_1(\mathbf{t}) = \{t_4, t_4, t_4, t_4, t_8, t_8, t_8, t_8, t_{12}, t_{12}, t_{12}, t_{12}\},$$

$$P_2(\mathbf{t}) = \{t_2, t_2, t_4, t_4, t_6, t_6, t_8, t_8, t_{10}, t_{10}, t_{12}, t_{12}\}.$$

. . 2. By working as in Lemma 1 we can see that:

$$(U^m)^{-1} = \left(\begin{array}{cc} H_m(\{1\}) & \left\{ H_{A(i+2)}\left(Q_{J(i)}(V_k^{p_{i+1}})\right), k = 1, \ldots, p_{i+1} - 1,\ i = 0, \ldots, N-1 \right\} \end{array} \right),$$

so the first column of $(U^m)^{-1}$ has only one nonzero entry $((U^m)^{-1})_{m,1} = 1$. All other columns result from dilation and translation of the column vector defined in (5); thus all columns of $(U^m)^{-1}$ have always two nonzero entries 1 and -1.

Finally, we establish a connection of the linear invertible transform: $T^m : \mathbf{C^m} \to \mathbf{C^m}$, $T^m(\mathbf{x}) = U^m\mathbf{x}$, with the DTT, which is a standard way in computer science and elsewhere for organizing information (see [7]). Recall that for any nonnegative collection of data $\mathbf{x} = \{x_r :\ r = 1, \ldots, m\}$, where m satisfies (1), the structure of DTT consists of N generations, with $J(n)$-branches in each generation, which we call walks:

$$a_{n,k} = \begin{cases} 0 & \text{if } R\left(n-1, \left\lceil \frac{k}{p_n} \right\rceil\right) = 0, \\ \frac{R(n,k)}{R\left(n-1, \left\lceil \frac{k}{p_n} \right\rceil\right)} & \text{if } R\left(n-1, \left\lceil \frac{k}{p_n} \right\rceil\right) \neq 0, \end{cases} \quad n = 1, \ldots N, k = 1, \ldots, J(n),$$

where $R(n,k)(\mathbf{x}) = \sum_{r=(k-1)A(n+1)+1}^{kA(n+1)} x_r$ and $J(n)$ and $A(n)$ are as in (1a) and (1b), respectively. The connection of the linear transform T^m with the DTT is obtained throughout the following estimation:

$$R(n,k) = \begin{cases} \left(T^m(\mathbf{x})\right)(1) & \text{if } n = 0, \\ \left(T^m(\mathbf{x})\right)\left(k - \lceil \frac{k}{p_n} \rceil + 2\right) & \text{if } k \notin [0]_{p_n} \text{ and } n = 1, \\ \left(T^m(\mathbf{x})\right)\left(J(n-1) + \lceil \frac{k}{p_n} \rceil + J(n-1)\mathrm{Mod}(k-1, p_n)\right) & \text{if } k \notin [0]_{p_n} \text{ and } n > 1, \\ R\left(n-1, \lceil \frac{k}{p_n} \rceil\right) - \sum_{i=k-p_n+1}^{k-1} R(n,i), & \text{if } k \in [0]_{p_n} \end{cases}$$

(8)

for $k = 1, \ldots, J(n)$.

Appendix A. Let $\mathbf{e}_i^{p_n}$ be row vectors of the identity matrix $\mathbf{I}_{p_n}$, and $V_j^{p_n}$, $j = 1, \ldots, p_n - 1$, are column matrices defined in (5); then

$$\begin{pmatrix} D_{p_n}(U^m(n-1)) \\ Q_{J(n-1)}\left(\mathbf{e}_1^{p_n}\right) \\ \vdots \\ Q_{J(n-1)}\left(\mathbf{e}_{p_n-1}^{p_n}\right) \end{pmatrix} \begin{pmatrix} Q_{J(n-1)}\left(\left(\mathbf{e}_{p_n}^{p_n}\right)^T\right) & Q_{J(n-1)}\left(V_1^{p_n}\right) & \ldots & Q_{J(n-1)}\left(V_{p_n-1}^{p_n}\right) \end{pmatrix}$$

$$= \begin{pmatrix} U^m(n-1) & & & \mathbf{O} \\ & \mathbf{I}_{J(n-1)} & & \\ & & \ddots & \\ \mathbf{O} & & & \mathbf{I}_{J(n-1)} \end{pmatrix}.$$

It suffices to prove that

(i) $D_{p_n}(U^m(n-1))Q_{J(n-1)}((\mathbf{e}_{p_n}^{p_n})^T) = U^m(n-1)$;

(ii) $D_{p_n}(U^m(n-1))Q_{J(n-1)}(V_j^{p_n}) = \mathbf{O},\ k = 1,\dots,p_n-1$, where $\mathbf{O}$ is the zero matrix in $\mathsf{M}_{J(n-1)}$;

(iii) $Q_{J(n-1)}(\mathbf{e}_k^{p_n})Q_{J(n-1)}((\mathbf{e}_{p_n}^{p_n})^T) = \mathbf{O},\ k = 1,\dots,p_n-1$, where $\mathbf{O}$ is the zero matrix in $\mathsf{M}_{J(n-1)}$;

(iv) $Q_{J(n-1)}(\mathbf{e}_j^{p_n})Q_{J(n-1)}(V_l^{p_n}) = \delta_{j,l}\mathbf{I}_{J(n-1)},\ j,l = 1,\dots,p_n-1$, and $\delta_{j,l}$ is Kronecker's delta.

Indeed we have the following:

(i) We use (2) and (3) to perform the following block-matrix multiplication:

$$D_{p_n}\left(U^m(n-1)\right)Q_{J(n-1)}\left(\left(\mathbf{e}_{p_n}^{p_n}\right)^T\right) = \left[D_{p_n}\left(\left(U^m(n-1)\right)_{i,j}\right)\left(\mathbf{e}_{p_n}^{p_n}\right)^T\right]_{i,j=1}^{J(n-1)} = U^m(n-1).$$

(ii) We observe that all column matrices $V_j^{p_n}$ have zero mean, so the block-matrix multiplication leads to:

$$D_{p_n}\left(U^m(n-1)\right)Q_{J(n-1)}\left(V_j^{p_n}\right) = \left[D_{p_n}\left(\left(U^m(n-1)\right)_{k,l}\right)V_j^{p_n}\right]_{k,l=1}^{J(n-1)}.$$

Since $D_{p_n}((U^m(n-1))_{k,l})V_j^{p_n} = (U^m(n-1))_{k,l}\sum_{r=1}^{p_n} v_{rj}^{p_n} = 0$, we have the result.

(iii) The obvious consequence of the fact that $\mathbf{e}_k^{p_n}\left(\mathbf{e}_{p_n}^{p_n}\right)^T = 0,\ k = 1,\dots,p_n-1$.

(iv) $Q_{J(n-1)}\left(\mathbf{e}_j^{p_n}\right)Q_{J(n-1)}\left(V_l^{p_n}\right) = \begin{pmatrix} \mathbf{e}_j^{p_n}V_l^{p_n} & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{e}_j^{p_n}V_l^{p_n} \end{pmatrix}$,

$j,l = 1,\dots,p_n-1$. Since $\mathbf{e}_j^{p_n}V_l^{p_n} = \sum_{k=1}^{p_n}\delta_{k,j}v_{k,l}^{p_n} = \delta_{j,l}$, we get the result. □

Appendix B. Let σ_n be the permutation defined in Proposition 1, then $\operatorname{sgn}\sigma_n = (-1)^{q_n}$, where q_n is the number of all inversions in the permutation σ_n and q_n equals to

$$q_n = \frac{p_n-1}{2}J(n-1)\left(1 + J(n-1) + \frac{p_n-2}{2}(J(n-1)-1)\right).$$

A pair of elements $(\sigma_n(i),\sigma_n(j))$ is called an inversion, if $i<j$ and $\sigma_n(i) > \sigma_n(j)$. The number of elements less than i to the right of i in σ_n gives the ith element of the inversion vector IV_{σ_n} corresponding to σ_n, and q_n equals the sum of all inversion vector elements.

The last column of the following matrix gives the elements of the inversion vector:

i	$\boldsymbol{\sigma_n}$(i)	**Inversion vector elements** $IV_{\sigma_n}(i)$
$1,\dots,J(n-1)$	ip_n	$i(p_n-1)$
$J(n-1)+1,\dots,2J(n-1)$	$1+\mathrm{Mod}(i-1,J(n-1))p_n$	$\mathrm{Mod}(i-1,J(n-1))(p_n-2)$
...	...	...
$(p_n-2)J(n-1)+1,\dots,(p_n-1)J(n-1)$	$p_n-2+\mathrm{Mod}(i-1,J(n-1))p_n$	$\mathrm{Mod}(i-1,J(n-1))$
$(p_n-1)J(n-1)+1,\dots,J(n)$	$p_n-1+\mathrm{Mod}(i-1,J(n-1))p_n$	0 for all i's

Now we have $\operatorname{sgn}\sigma_n = (-1)^{q_n}$, where

$$q_n = \sum_{i=1}^{J(n)} IV_{\sigma_n}(i) = \sum_{i=1}^{J(n-1)} i(p_n-1) + \sum_{i=J(n-1)+1}^{2J(n-1)} \operatorname{Mod}(i-1, J(n-1))(p_n-2) + \cdots$$

$$= (p_n-1)\sum_{i=1}^{J(n-1)} i + (p_n-2)\sum_{i=1}^{J(n-1)-1} i + \cdots + \sum_{i=1}^{J(n-1)-1} i$$

$$= (p_n-1)\frac{J(n-1)\,(1+J(n-1))}{2} + \frac{J(n-1)\,(J(n-1)-1)}{2}\,\frac{(p_n-2)(p_n-1)}{2},$$

and elementary calculations yield the result. □

Acknowledgment. The authors thank the unknown referee for valuable remarks concerning the notation and comments about the reorganization of the text and the exposition of proofs.

REFERENCES

[1] N. Atreas and C. Karanikas, *Sampling formulas for spectral wavelet analysis on spaces of p^M-periodic sequences for computational applications on edge detection*, Numer. Funct. Anal. Optim., 26 (2005), pp. 285–301.

[2] M. W. Berry, Z. Drmac, and E. R. Jessup, *Matrices, vector spaces, and information retrieval*, SIAM Rev., 41 (1999), pp. 335–362.

[3] N. D. Atreas, C. Karanikas, and A. O. Tarakanov, *Signal Processing by an Immune Type Tree Transform*, Lect. Notes in Comput. Sci. 2787, J. Timmis et al., eds., Springer-Verlag, Berlin, Heidelberg, pp. 111–119.

[4] J. Dongarra, *Sparse matrix storage formats*, in Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, Z. Bai et al., eds., SIAM, Philadelphia, 2000; also available online from http://www.cd.utk.edu/∼dongarra/etemplates/node372.html.

[5] M. Holschneider, *Wavelets an Analysis Tool*, Clarendon Press, Oxford, 1995.

[6] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[7] C. Karanikas and G. Proios, *A discrete transform based on the tree structure of a data for pattern recognition of immune type*, Chaos Solitions Fractals, 17 (2003), pp. 195–201.

[8] E. Montagne and A. Ekambaram, *An optimal storage format for sparse matrices*, Inform. Process. Lett., 90 (2004), pp. 87–92.

[9] G. G. Walter, *A sampling theorem for wavelet subspaces*, IEEE Trans. Inform. Theory, 38 (1992), pp. 881–884.

[10] A. I. Zayed, *Advances in Shannon's Sampling Theory*, CRC Press, Boca Raton, FL, 1993.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 324–345

OPTIMIZING THE COUPLING BETWEEN TWO ISOMETRIC PROJECTIONS OF MATRICES*

CATHERINE FRAIKIN†, YURII NESTEROV‡, AND PAUL VAN DOOREN†

Abstract. In this paper, we analyze the coupling between the isometric projections of two square matrices. These two matrices of dimensions $m \times m$ and $n \times n$ are restricted to a lower k-dimensional subspace under isometry constraints. We maximize the coupling between these isometric projections expressed as the trace of the product of the projected matrices. First we connect this problem to notions such as the generalized numerical range, the field of values, and the similarity matrix. We show that these concepts are particular cases of our problem for special choices of m, n, and k. The formulation used here applies to both real and complex matrices. We characterize the objective function, its critical points, and its optimal value for Hermitian and normal matrices, and, finally, give upper and lower bounds for the general case. An iterative algorithm based on the singular value decomposition is proposed to solve the optimization problem.

Key words. trace maximization, generalized numerical range, isometry, singular value decompostion

AMS subject classifications. 15A60, 47A12, 65F30, 65K10

DOI. 10.1137/050643878

1. Introduction.

The problem of projection of matrices in lower-dimensional subspaces is of great interest for a large field of applications. The projection of matrices provides an easier visualization and comprehension of the initial problem and is often used to reduce the complexity of some computational problems. Moreover the coupling between these projections can reveal some particularities inherent to the data which can be analyzed and interpreted.

We consider the coupling or similarity between two "projected" matrices A and B, respectively, of dimensions $m \times m$ and $n \times n$, expressed as the real part of the trace of the product of the isometric projections U^*AU and V^*BV:

$$\Re \operatorname{tr}(U^*AUV^*B^*V) \tag{1.1}$$

under the constraint that $U^*U = V^*V = I_k$, where I_k denotes the identity matrix of dimension k, with $k \leq \min(m, n)$. In this paper, we will consider both real and complex matrices. The notation will be different for the real and complex cases, i.e., ^{T} and *, respectively for the transpose and complex conjugate transpose, the real inner product and real-valued inner product, respectively, for the real and complex case (see the notations in section 2.1). In particular, for real matrices, the coupling we consider is the following:

$$\operatorname{tr}(U^TAUV^TB^TV). \tag{1.2}$$

*Received by the editors October 31, 2005; accepted for publication (in revised form) by M. L. Overton October 16, 2007; published electronically March 19, 2008. This paper presents research results of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office, and a grant Action de Recherche Concertée (ARC) of the Communauté Française de Belgique. The scientific responsibility rests with its authors.

http://www.siam.org/journals/simax/30-1/64387.html

†Department of Mathematical Engineering, Université catholique de Louvain (UCL), Bâtiment Euler, 4 avenue Georges Lemaître, B-1348 Louvain-La-Neuve, Belgium (fraikin@inma.ucl.ac.be, vdooren@inma.ucl.ac.be).

‡Bâtiment CORE, 34 voie du Roman Pays, B-1348 Louvain-La-Neuve, Belgium (nesterov@inma.ucl.ac.be).

Most results are developed for the complex case. When results are different for the real and complex problems, we explicitly mention this; otherwise, we consider only the complex problem.

This is a generic problem which can be linked to various applications treated in the literature and which has been studied extensively in a variety of contexts for particular dimensions of the projection and of the matrices. A first field of application for real matrices lies in the analysis of graphs. The notion of the graph similarity matrix, which is a matrix that expresses how similar the nodes of two graphs are, has recently been introduced in [2]. For undirected graphs, the similarity matrix is the product of the isometries U and V^T maximizing (1.2) where A and B are real symmetric adjacency matrices and which is obtained with k equal to one. The graph similarity matrix is, e.g., useful for the development of efficient search engines or the automatic extraction of synonyms in a dictionary. Another important task in graph analysis is that of graph matching, which is a fundamental problem in pattern recognition and in shape and image analysis (see, e.g., [6] for an overview of graph matching techniques). A common class of methods in graph matching is the spectral methods in which spectral properties of characteristic matrices are used to compare the graphs. The spectral method developed in [4] combines a projection technique and a clustering algorithm to match the graphs in a lower-dimensional subspace. It can be shown that the step of projection used by the authors is equivalent to maximizing (1.2) for symmetric matrices A and B. A second field of application where relevant matrices are complex concerns experiments in quantum mechanics and in particular the task of maximizing the signal intensity in coherent ensemble spectroscopy (see, e.g., [7], [9], [16]). Indeed, the spectroscopic experiments require optimal unitary transformations of a given initial operator onto a target operator maximizing the overlap between these two operators. From a mathematical point of view, maximizing this overlap is equivalent to maximizing an expression similar to (1.1) where all of the matrices are square. The optimal value constitutes a transfer bound called the generalized numerical radius of A.

In the linear algebra literature, problem (1.1) has also been studied for particular cases and dimensions, and it hence constitutes an extension of a variety of known problems. For the case where all of the matrices are square, this problem corresponds to the generalized numerical range. See, e.g., [13] or [14] for a survey on the properties of the generalized numerical range. For the scalar case, which corresponds to a one-dimensional projection, the expression (1.1) is equivalent to the product of the field of values of two matrices (see, e.g., [12]). In this paper we consider matrices A and B of different dimensions and an arbitrary dimension k. We treat also the complex and real cases.

There exist many numerical algorithms to maximize (1.1) for particular dimensions of the matrices (e.g., [1], [3], [7], [9]). We develop here a simple recursive algorithm valid for the general case, i.e., for complex or real problems and for all dimensions of the matrices. Characterizations of the fixed points of the algorithm are presented.

The paper is organized as follows. In section 2, we introduce some notations. In section 3, we define the problem considered in the paper which consists of maximizing an expression similar to (1.1). We recall some important results from the literature that we can link to our problem. The first one concerns square matrices and appears in the field of the generalized numerical range and in the context of semidefinite programming relaxations. The second case is about one-dimensional projections and is linked to the field of values of matrices. We extend also some of these results. The

main new results are in section 4, where we characterize the critical points of the problem. Then we focus on the case of Hermitian and normal matrices, and we give lower and upper bounds for the optimal value. In section 5, we propose a simple algorithm to solve the optimization problem. Some numerical experiments are also presented. The last section 6 summarizes the results and describes some directions for future research.

2. Notations. In this section, we introduce some notations used in the paper. The first part treats the complex and real-valued inner product of matrices. The second part summarizes some definitions and results about gradients of functions with matrix arguments. Finally, the definitions of an isometry and of an isometric projection are given.

2.1. Inner product. Let $\mathbb{R}^{m\times n}$ and $\mathbb{C}^{m\times n}$ denote the set of all $m \times n$ real and complex matrices, respectively, and let X^T, $\bar{X}$, and X^* represent, respectively, the transpose, the complex conjugate, and the complex conjugate transpose of X. The inner product between matrices is defined as follows. For X, $Y \in \mathbb{R}^{m\times n}$, the [illegible] is denoted by

$$\langle X, Y\rangle = \sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}Y_{ij} \tag{2.1}$$

and can be linked to the trace function of a matrix:

$$\langle X, Y\rangle = \operatorname{tr}(XY^T) = \operatorname{tr}(X^TY).$$

For complex matrices X, $Y \in \mathbb{C}^{m\times n}$, another inner product often useful in optimization is the [illegible] defined by:

$$\langle X, Y\rangle_H = \langle \Re(X), \Re(Y)\rangle + \langle \Im(X), \Im(Y)\rangle, \tag{2.2}$$

where $\Re(X)$ and $\Im(X)$ represent the real and the imaginary part of X, respectively. This inner product can be linked again to the trace

$$\langle X, Y\rangle_H = \Re \operatorname{tr}(X^*Y)$$

and satisfies the following properties:

$$\langle X, Y\rangle_H = \langle Y^*, X^*\rangle_H = \overline{\langle X^*, Y^*\rangle_H}. \tag{2.3}$$

For complex vectors x, $y \in \mathbb{C}^n$, the real-valued inner product is defined similarly by:

$$\langle x, y\rangle_H = \sum_{i=1}^{n} \Re(\bar{x}_i y_i) = \langle \Re(x), \Re(y)\rangle + \langle \Im(x), \Im(y)\rangle. \tag{2.4}$$

2.2. Gradients. Let $f : \mathbb{C}^{m\times n} \to \mathbb{R}$ be a differentiable real-valued function with matrix argument X. Then the first-order approximation of f at a point X can be expressed as

$$f(X + \Delta) = f(X) + \langle \nabla f(X), \Delta\rangle_H + o(\|\Delta\|), \tag{2.5}$$

where the gradient $\nabla f(X)$ is the $m\times n$ matrix whose (i, j) entry is $\frac{\partial f(X)}{\partial X_{i,j}}$. As particular examples, we provide some gradients of inner-product functions with respect to a matrix X:

$$\nabla\langle A, X^*X\rangle_H = X(A + A^*), \tag{2.6}$$

$$\nabla\langle X^*AX, B\rangle_H = AXB^* + A^*XB. \tag{2.7}$$

2.3. Isometry and isometric projection. Let $A \in \mathbb{C}^{m\times m}$ and $U \in \mathbb{C}^{m\times k}$, $k \leq m$, be given. If $U^*U = I_k$, with I_k the identity matrix of dimension k, then U is called an [illegible] and U^*AU is called an [illegible] of A.

3. Main known results and some extensions of these results. For $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, we consider the following problem:

$$\max_{\substack{U^*U=I_k\\ V^*V=I_k}} \langle U^*AU, V^*BV\rangle_H, \tag{3.1}$$

where $U \in \mathbb{C}^{m\times k}$ and $V \in \mathbb{C}^{n\times k}$, with $k \leq \min(m,n)$. If the matrices A and B are real, one could restrict U and V to be real also which is then a different problem expressed by:

$$\max_{\substack{U^TU=I_k\\ V^TV=I_k}} \langle U^TAU, V^TBV\rangle. \tag{3.2}$$

It will be clear, depending on the context, which case we consider. Since the algebraic structure of the constraints and the objective function is the same, most results for both problems will be essentially the same. Let us remark that, for $k = \min(m,n)$, (3.1) is equivalent to

$$\max_{Q^*Q=I_n} \langle Q^*AQ, B\rangle_H, \tag{3.3}$$

where $Q = UV^*$ is an isometry of dimension $m \times n$. The general problem is then reduced to an optimization problem over only one variable Q.

This problem has largely been studied for particular dimensions of m, n, and k. Section 3.1 contains results for $k = m = n$, while section 3.2 summarizes some properties for $k = 1$.

3.1. Square matrices U and V. In the case where m, n, and k are equal, U and V are square matrices, and the problem is reduced to (3.3). This problem has been studied in a variety of contexts. In the rest of the section, we summarize some important results for the generalized numerical range (or C-numerical range) and for semidefinite programming relaxations providing bounds on the problem. To link the notations used in the literature for this problem with (3.3), we point out that

$$\langle Q^*AQ, B\rangle_H = \Re\left(\operatorname{tr}(AQB^*Q^*)\right) = \Re\left(\operatorname{tr}(B^*Q^*AQ)\right).$$

3.1.1. C-numerical range. The problem (3.3) is equivalent to maximizing the real part of the C-numerical range of A (or generalized numerical range) introduced by [8] and defined by

$$W_C(A) := \{\operatorname{tr}(C^*Q^*AQ) : Q \text{ is unitary}\}. \tag{3.4}$$

See, e.g., [13] for a survey on the properties of the C-numerical range. In the literature it is pointed out that the C-numerical range and in particular its geometry can be quite complicated. For all $A \in \mathbb{C}^{n\times n}$, $W_C(A)$ is convex if C is Hermitian or if C is normal with its eigenvalues colinear in the complex plane. Moreover, for general A and C, $W_C(A)$ is always star-shaped with respect to the star-center $(\operatorname{tr} A)(\operatorname{tr} C)/n$ [5] but not necessarily convex. For example, [19] gave an example in which C is normal but not Hermitian and where $W_C(A)$ is not convex. Upper bounds on the size of

$W_C(A)$ are given in [9]. In the general case there is no closed formula for computing the C-numerical range. One can only come up with an approximation. For example, [9] provides a gradient flow leading to a numerical algorithm to approach the set of critical points of $\Re(\mathrm{tr}(C^*Q^*AQ))$. See also [3] and [7].

The C-numerical range has been studied by many authors in the past few decades and has many domains of applications, e.g., in quantum dynamics for the study of the efficiency of polarization or coherence transfer between quantized states under unitary transformations. This application is equivalent to computing the C-numerical radius of A for certain sparse nilpotent matrices C and A (e.g., [7], [9], [16]). Some authors have also used the numerical range to study problems on norms of operators (see, e.g., [15]).

3.1.2. Semidefinite programming relaxations.

In the case of real and symmetric matrices A and B, the problem (3.3) is reduced to a classical problem called the quadratically constrained quadratic program defined over orthogonal matrices:

$$\mu_P = \max_{Q^TQ=I} \mathrm{tr}(AQBQ^T). \tag{3.5}$$

This problem can be solved exactly, and the optimal value is obtained by performing spectral decompositions of A and B (see, e.g., [1] or [18]). Let us suppose that the orthogonal diagonalizations of A, B are $A = UD_AU^T$ and $B = VD_BV^T$, respectively, where the eigenvalues in D_A and in D_B are ordered in a nondecreasing fashion. Then the optimal value of (3.5) is $\mathrm{tr}\, D_AD_B$, and the optimal solution is obtained by using the orthogonal matrices that yield the diagonalizations, i.e., $Q_{opt} = UV^T$.

For real matrices and by a reasoning similar to the one developed in [20], we construct the following primal problem ν_P and its semidefinite programming relaxation ν_D:

$$\begin{aligned}
&\nu_P = \max_{\substack{Q^TQ=I\\ QQ^T=I}} \mathrm{tr}(AQB^TQ^T),\\
&\nu_D = \min\ \mathrm{tr}\, S + \mathrm{tr}\, T \qquad (3.6)\\
&\quad \text{such that (s.t.)} \quad \frac{B\otimes A}{2} + \frac{B^T\otimes A^T}{2} - S\otimes I - I\otimes T \preceq 0,\\
&\qquad\qquad\qquad\quad S = S^T,\\
&\qquad\qquad\qquad\quad T = T^T,
\end{aligned}$$

where S and T are the symmetric matrices of Lagrange multipliers used to relax the constraints $Q^TQ = I$ and $QQ^T = I$ and $\otimes$ denotes the Kronecker product. The redundant constraint $QQ^T = I$ is added in order to close the duality gap for symmetric matrices A and B. Indeed, for symmetric A and B, it is proved that strong duality holds; $\nu_P = \nu_D$ [1]. A few examples show that there can be a nonzero duality gap in the case of arbitrary matrices which are not symmetric. Strong duality does not hold in this case, but this semidefinite relaxation provides an upper bound ν_D for the problem we consider $\nu_P \leq \nu_D$. See section 5.3.2 for an example where a duality gap occurs.

A complex matrix $A = A_R + jA_I$, with $j = \sqrt{-1}$, of dimension $n \times n$ can be represented by a real matrix $\tilde{A}$ of dimension $2n \times 2n$ of the form:

$$\tilde{A} = \begin{pmatrix} A_R & A_I \\ -A_I & A_R \end{pmatrix}. \tag{3.7}$$

For a Hermitian matrix A, $\tilde{A}$ is symmetric, while, for a unitary matrix Q, $\tilde{Q}$ is orthogonal. If the matrices A, B, and $Q \in \mathbb{C}^{n\times n}$ are represented by the matrices $\tilde{A}$, $\tilde{B}$, and $\tilde{Q} \in \mathbb{C}^{2n\times 2n}$, respectively, we obtain the following link between the two trace functions:

$$2\Re \operatorname{tr}(AQB^*Q^*) = \operatorname{tr}(\tilde{A}\tilde{Q}\tilde{B}^T\tilde{Q}^T). \tag{3.8}$$

One can easily prove this relation by developing the two terms of the equality. The problem (3.5) in terms of complex matrices is thus equivalent to maximizing

$$\max_{\tilde{Q}^T\tilde{Q}=I} \frac{1}{2} \operatorname{tr}(\tilde{A}\tilde{Q}\tilde{B}^T\tilde{Q}^T) \tag{3.9}$$

expressed in terms of real matrices. The dual method developed previously for real matrices can then be applied in the same way and provides an upper bound for the problem.

For Hermitian matrices A and B, strong duality holds because the representations $\tilde{A}$ and $\tilde{B}$ are symmetric, and then the gap between the primal and dual problems is zero. In this case, we know that the solution is simply the trace of the product of the diagonal matrices of the eigenvalues of $\tilde{A}$ and $\tilde{B}$ ordered in an adequate way $\operatorname{tr} D_{\tilde{A}} D_{\tilde{B}}$. By developing the expressions one can easily see that $\operatorname{tr} D_{\tilde{A}} D_{\tilde{B}} = 2 \operatorname{tr} D_A D_B$. The optimal value obtained for Hermitian matrices is then the product of the eigenvalues of the matrices. For general complex matrices A and B, the dual problem provides only an upper bound for the initial problem.

3.2. The one-dimensional case. When k equals one, the matrices U and V are reduced to vectors u and v, respectively, and the problem (3.1) becomes

$$\max_{\substack{u^*u=1\\ v^*v=1}} \langle u^*Au, v^*Bv\rangle_H. \tag{3.10}$$

This problem is related to the notion of the field of values. The field of values of a matrix A (also known as the numerical range) is defined by [12]

$$F(A) := \{x^*Ax : x \in \mathbb{C}^n,\, x^*x = 1\}.$$

The problem is then reduced to obtaining the maximum of the products of the elements from the fields of values of A and of B.

The field of values is known to be a convex subset of the complex plane, while the product of two fields of values $F(A)F(B)$ is generally not a convex set. We provide a simple counterexample.

1. Let

$$A = \begin{pmatrix} 1 & 0 \\ 0 & j \end{pmatrix}, \quad B = \begin{pmatrix} -1 & 0 \\ 0 & -j \end{pmatrix}.$$

Then $F(A)$ is the line segment joining 1 and j, and $F(B)$ is the line segment joining -1 and $-j$. Thus $F(A)F(B)$ is not a convex set since $1, -1 \in F(A)F(B)$ and $0 \notin F(A)F(B)$.

In the real and Hermitian cases, we obtain the exact optimal value of the function.

3.2.1. Hermitian case. For a Hermitian matrix A, the field of values is the interval

$$[\lambda_{min}(A), \lambda_{max}(A)],$$

with $\lambda_{min}(A)$ and $\lambda_{max}(A)$ the smallest and largest eigenvalues of A, respectively. The solution of (3.10) is then the product of the adequate extremal (smallest and largest) eigenvalues of the Hermitian matrices A and B depending on their signs. The solutions u and v providing the optimum are the eigenvectors of A and B corresponding to the eigenvalues providing the solution, respectively.

3.2.2. Real case. For a real matrix A, the field of values could be complex in general. The , field of values associated with a real square matrix A is defined by [12]

$$F_R(A) := \{x^T A x : x \in \mathbb{R}^n, x^T x = 1\}.$$

If we notice that $F_R(A) = F_R(A_H)$, with $A_H = \frac{A+A^T}{2}$ the symmetric part of A, then it is sufficient to consider only the symmetric part of the matrix in order to study the real field of values. $F_R(A)$ is the real interval joining the smallest and the largest eigenvalues of A_H and is thus always convex. The solution of (3.10), for A, B, u, and v real, is then the product of the adequate extremal eigenvalues of A_H and B_H depending on their signs.

In the particular case of real symmetric matrices, this scalar case can be linked to the concept of the similarity matrix S introduced in [2]. This matrix expresses how similar vertices of two graphs are and is defined as a particular fixed point of the iteration

$$S_{k+1} = \frac{AS_kB^T + A^TS_kB}{\|AS_kB^T + A^TS_kB\|_F}, \tag{3.11}$$

with the Frobenius norm $\|.\|_F = \sqrt{\langle .,. \rangle}$ and where A and B, representing the adjacency matrices of the graphs, have nonnegative elements. In the case where the adjacency matrix of one graph is normal, the similarity matrix has rank one and can then be decomposed into the product of two vectors u and v, $S = uv^T$, and it satisfies the equation $\rho S = ASB^T + A^TSB$ [2]. In the case of undirected graphs which are characterized by symmetric adjacency matrices, u and v are the Perron vectors of A and B. The solutions u and v of (3.10) are then those giving the similarity matrix S. In general S is not of rank one, but we will see in section 5.1 an algorithm to solve the corresponding optimization problem. The similarity matrix can be linked to our problem and is obtained as the limit of the normalized iterates $Au_iv_i^TB^T + A^Tu_iv_i^TB$.

4. The general case. In this section we provide some results obtained for the general problem

$$\max_{\substack{U^*U=I_k \\ V^*V=I_k}} \langle U^*AU, V^*BV \rangle_H, \tag{4.1}$$

where $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $U \in \mathbb{C}^{m\times k}$, and $V \in \mathbb{C}^{n\times k}$, with $k \leq \min(m,n)$.

We derive first the expressions for the critical points of the optimization problem. Then we consider some particular cases, i.e., when one matrix is Hermitian and when the two matrices are normal. An upper and a lower bound to the general problem are also obtained by decomposing the problem into the sum of two Hermitian problems. Let us mention that the techniques used in the rest of the section are quite similar to the ones used in [9].

4.1. Critical points. We consider the following problem:

$$\max_{\substack{U^*U=I_k \\ V^*V=I_k}} F(U,V), \tag{4.2}$$

where the objective function

$$F(U,V) = \langle U^*AU, V^*BV\rangle_H = \frac{1}{2}\left[\langle U^*AU, V^*BV\rangle_H + \langle U^*A^*U, V^*B^*V\rangle_H\right] \tag{4.3}$$

according to the trace properties (2.3). The problem is equivalent to maximizing the coupling between two constrained matrices A and B. This is an optimization problem of a continuous function $F(U,V)$ on a compact domain. In particular, the constraint set constitutes a smooth manifold as a product of two compact Stiefel manifolds (see, e.g., [10]). There always exists a solution U and V optimizing the function such that the first-order conditions are satisfied.

The first-order optimality conditions for (4.2) can be derived from the Lagrangian $L(U,V,X,Y)$

$$\begin{aligned} L(U,V,X,Y) = \frac{1}{2}[&\langle U^*AU, V^*BV\rangle_H + \langle U^*A^*U, V^*B^*V\rangle_H \\ &+ \langle X, I - U^*U\rangle_H + \langle Y, I - V^*V\rangle_H], \end{aligned} \tag{4.4}$$

where X and Y are Hermitian matrices of Lagrange multipliers for the isometry constraints. Partial gradients of L with respect to (U,V) according to (2.6) and (2.7) lead to the following first-order optimality conditions:

$$\begin{aligned} \nabla_U L &= AU\,(V^*B^*V) + A^*U\,(V^*BV) - UX = 0, \\ \nabla_V L &= BV\,(U^*A^*U) + B^*V\,(U^*AU) - VY = 0 \end{aligned}$$

or, equivalently,

$$\begin{aligned} UX &= AU(V^*B^*V) + A^*U(V^*BV), \\ VY &= BV(U^*A^*U) + B^*V(U^*AU), \end{aligned}$$

and of course the constraints $U^*U = V^*V = I$. It easily follows from this that $X = Y$. If we decompose $X = Y$ by an eigendecomposition $\hat{U}\Lambda\hat{U}^*$, where $\hat{U} \in \mathbb{C}^{k\times k}$ is an unitary matrix, then we can replace U by $U\hat{U}$ and V by $V\hat{U}$ which amounts to changing the bases in which we describe the spaces $Im(U)$ and $Im(V)$, the images of U and V. In this particular coordinate system the above first-order conditions would have a real diagonal matrix Λ with ordered diagonal elements $\lambda_i \geq \lambda_{i+1}$, $i = 1,\ldots,k-1$:

$$\begin{aligned} U\Lambda &= AU(V^*B^*V) + A^*U(V^*BV), \\ V\Lambda &= BV(U^*A^*U) + B^*V(U^*AU). \end{aligned} \tag{4.5}$$

4.2. Case where one matrix is Hermitian. If A is Hermitian or $A = A^*$, the maximum of (4.1) is achieved for matrices U and V corresponding, respectively, to the dominant eigenvectors of A and $(B + B^*)$. Moreover $U\Lambda V^*$ is exactly of rank k. In other words, in this case the problem is decoupled regarding the matrices A and B, and the solutions U and V satisfy

$$\begin{aligned} AU &= UA_{sub}, \\ (B + B^*)V &= VB_{sub}, \end{aligned} \tag{4.6}$$

where A_{sub} and B_{sub} are diagonal matrices of dimension k whose elements are the dominant eigenvalues of A and $(B+B^*)$, respectively. The following theorem provides a characterization of the maximum of (4.1).

THEOREM 4.1. , $A = A^*$,

$$\max_{\substack{U^*U=I_k \\ V^*V=I_k}} \langle U^*AU, V^*BV\rangle_H = \frac{1}{2}\max_{\pi_1,\pi_2}\left(\sum_{i=1}^{k}\alpha_{\pi_1(i)}\beta_{\pi_2(i)}\right), \tag{4.7}$$

, $\alpha_1,\ldots,\alpha_m$, $\beta_1,\ldots,\beta_n$, , , , , A , $(B+B^*)$, $\pi_1(.)$, $\pi_2(.)$, , , , , $1,\ldots,m$, $1,\ldots n$, , , , . The solution (U,V) satisfies the equations for the critical points

$$\begin{aligned} \Lambda &= U^*AUV^*(B+B^*)V, \\ \Lambda &= V^*(B+B^*)VU^*AU, \end{aligned} \tag{4.8}$$

which point out that the two matrices U^*AU and $V^*(B+B^*)V$ commute and are thus simultaneously diagonalizable under the same unitary transformation W such that

$$\begin{aligned} W^*U^*AUW &= D_A, \\ W^*V^*(B+B^*)VW &= D_B. \end{aligned} \tag{4.9}$$

The expressions in (4.8) become

$$\begin{aligned} W^*\Lambda W = \hat{\Lambda} &= D_AD_B, \\ W^*\Lambda W = \hat{\Lambda} &= D_BD_A, \end{aligned} \tag{4.10}$$

where $\hat{\Lambda}$ is diagonal as a product of two diagonal matrices D_A and D_B. In this coordinate system, the critical point condition (4.5) can be expressed as

$$\begin{aligned} u_i\lambda_i &= Au_i\beta_i, \\ v_i\lambda_i &= (B+B^*)v_i\alpha_i, \end{aligned}$$

where α_i and β_i are the eigenvalues of A and $(B+B^*)$, respectively. If $\lambda_i \neq 0$, it is obvious that u_i and v_i are eigenvectors of A and $(B+B^*)$, respectively. The matrices U and V providing the optimum are thus composed of the dominant eigenvectors. If $\lambda_i = 0$, the above formulas do not imply that both u_i and v_i are eigenvectors, since only one of α_i and β_i needs to be zero, but it easy to see that one can choose both u_i and v_i to be eigenvectors without altering the objective function.

It follows from (4.8) and (4.10) that the value of F for a critical point is equal to the trace of

$$\frac{1}{2}W^*U^*AUV^*(B+B^*)VW. \tag{4.11}$$

The maximal value for all of the critical points is therefore obtained by

$$\frac{1}{2}\max_{\pi_1,\pi_2}\left(\sum_{i=1}^{k}\alpha_{\pi_1(i)}\beta_{\pi_2(i)}\right), \tag{4.12}$$

where $\alpha_1,\ldots,\alpha_m$ and $\beta_1,\ldots,\beta_n$ are the real eigenvalues of A and $(B+B^*)$ and $\pi_1(.)$ and $\pi_2(.)$ permutations of $1,\ldots,m$ and $1,\ldots n$, respectively. □

Let us remark that, in the square case ($k = m = n$), Theorem 4.1 is a well-known fact (see, e.g., [1]). In practice, problem (4.12) can be solved by the following theorem (see Figure 4.1 for the notations).

LEMMA 4.2. [illegible] $\{\alpha_i, i = 1, \dots, m\}$ [illegible] $\{\beta_i, i = 1, \dots, n\}$ [illegible] A [illegible] $(B + B^*)$ [illegible] m_+ [illegible] m_- [illegible] ([illegible]) [illegible] A [illegible] n_+ [illegible] n_- [illegible] $(B + B^*)$ [illegible] $k_+ = \min(m_+, n_+)$, $k_- = \min(m_-, n_-)$ [illegible] $l = k - (k_+ + k_-)$

[illegible] $k \le k_+ + k_-$ [illegible] (4.12) [illegible] k_+ [illegible] $\{(\alpha_i, \beta_i), i = 1, \dots, k_+\}$ [illegible] k_- [illegible] $\{(\alpha_{m-i+1}, \beta_{n-i+1}), i = 1, \dots, k_-\}$ [illegible] k [illegible]

[illegible] $k > k_+ + k_-$ [illegible] $k_+ + k_-$ [illegible] l [illegible]

$$\begin{cases} (\alpha_{m_+-i+1}, \beta_{n_++l-i+1}), i = 1, \dots, l & \text{if } k_- = m_-, \\ (\alpha_{m_++l-i+1}, \beta_{n_+-i+1}), i = 1, \dots, l & \text{if } k_- = n_-. \end{cases} \tag{4.13}$$

[illegible]. Let us put the points $\{\alpha_i, i = 1, \dots, m\}$ and $\{\beta_i, i = 1, \dots, n\}$ on two parallel axes and connect the elements of the selected couples by a line (see Figure 4.1). The optimal couples satisfy two properties.

1. The elements have to be coupled such that no crossing between the pairs appears. In other words, only parallel couplings are allowed. Indeed, if we consider the couples (α_p, β_q) and (α_r, β_s), with $p > r$, $q > s$, i.e., $\alpha_p \le \alpha_r$ and $\beta_q \le \beta_s$, we have

$$\alpha_p\beta_q + \alpha_r\beta_s - \alpha_p\beta_s - \alpha_r\beta_q = (\alpha_p - \alpha_r)(\beta_q - \beta_s) \ge 0.$$

The combination of the two couples (α_p, β_q) and (α_r, β_s) produces therefore a larger value than the value obtained for (α_p, β_s) and (α_r, β_q).

2. The pairs formed by elements of the same sign have to be chosen first, since their product is nonnegative and therefore larger than a product of two elements of different sign.

Following these two properties, one can easily see that, for $k \le k_+ + k_-$, the couples of eigenvalues producing (4.12) are the couples formed by the extremal eigenvalues of A and $(B + B^*)$. Indeed these products are all nonnegative and maximize the function.

For $k > k_+ + k_-$, we take into account as well negative products that have to be as small as possible in absolute value for all of the combinations of eigenvalues. Two cases may occur: $k_- = m_-$ or $k_- = n_-$. We consider $k_- = m_-$ (see Figure 4.1), and the reasoning for $k_- = n_-$ is similar. In this case, the first $k_+ + k_-$ pairs are the pairs formed by elements of the same sign according to the second property. The l remaining couples are formed as expressed by (4.13), which takes into account the first property of no crossing between the elements. This expression also takes into account that only the elements closest to zero are kept. Indeed, permuting any element from this set with an element farther from zero will give a product, of negative value, greater in absolute value. For example, if we take α_{m_+-3} instead of α_{m_+-2} in Figure 4.1, we obtain the couple $\alpha_{m_+-3}\beta_{n_++1}$, which is smaller than $\alpha_{m_+-2}\beta_{n_++1}$. By reasoning similarly for all of the elements, one proves that only the elements closest to zero yield (4.12). □

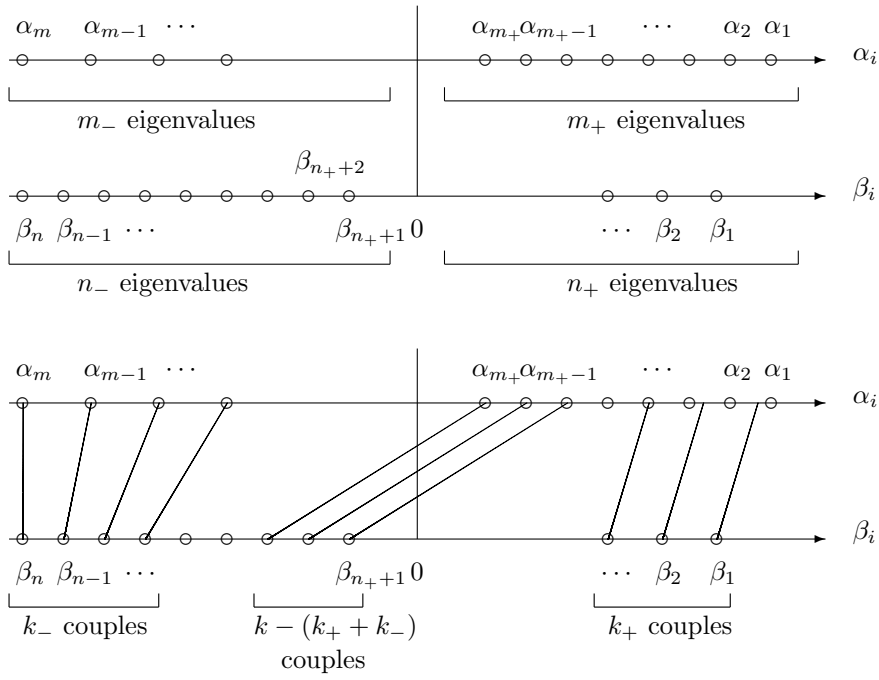


FIG. 4.1. *Representation of the eigenvalues α_i of A and β_i of B with their numbering (on the top) and of the k adequate couples for finding the maximal combination* (4.12) *(on the bottom).*

4.3. Sum of two Hermitian problems. The square matrices A and B can always be decomposed into

$$A = A_H + jA_S, \quad B = B_H + jB_S,$$

where the matrices $A_H = \frac{A+A^*}{2}$ and $A_S = \frac{A-A^*}{2j}$ are Hermitian matrices. The objective function can also then be decomposed into a sum of two Hermitian problems

$$\frac{1}{2}\langle U^*A_HU, V^*B_HV\rangle_H + \frac{1}{2}\langle U^*A_SU, V^*B_SV\rangle_H. \tag{4.14}$$

This expression of the objective function provides an upper bound for the optimal value. This bound is expressed in the following corollary.

COROLLARY 4.3. [illegible] 4.1 [illegible] (4.2) [illegible] $F(U,V)$ [illegible] (4.14)[illegible]

$$\frac{1}{2}\max_{\pi_1,\pi_2}\left(\sum_{i=1}^{k}\alpha^H_{\pi_1(i)}\beta^H_{\pi_2(i)}\right) + \frac{1}{2}\max_{\pi_1,\pi_2}\left(\sum_{i=1}^{k}\alpha^S_{\pi_1(i)}\beta^S_{\pi_2(i)}\right), \tag{4.15}$$

[illegible] α^H_i [illegible] β^H_i [illegible] A_H [illegible] B_H [illegible] α^S_j [illegible] β^S_j [illegible] A_S [illegible] B_S [illegible] $\pi_1(.)$ [illegible] $\pi_2(.)$ [illegible] $1,\ldots,m$ [illegible] $1,\ldots n$ [illegible]

Let us point out that the permutations π_1 and π_2 in these two terms at the respective maxima can be different. A lower bound can also be found by choosing the matrices U and V optimizing one of the two Hermitian problems and by calculating the value of (4.14) for this pair of matrices (U,V). For example, if we take U_1, V_1 optimum

for $\frac{1}{2}\langle U^*A_HU, V^*B_HV\rangle_H$, the optimal solution is lower and upper bounded by

$$\begin{aligned}
&\frac{1}{2}\max_{\pi_1,\pi_2}\left(\sum_{i=1}^{k}\alpha^H_{\pi_1(i)}\beta^H_{\pi_2(i)}\right)+\frac{1}{2}\langle U_1^*A_SU_1, V_1^*B_SV_1\rangle_H \\
&\le \max_{\substack{U^*U=I_k\\ V^*V=I_k}} \frac{1}{2}\langle U^*A_HU, V^*B_HV\rangle_H+\frac{1}{2}\langle U^*A_SU, V^*B_SV\rangle_H \qquad (4.16)\\
&\le \frac{1}{2}\max_{\pi_1,\pi_2}\left(\sum_{i=1}^{k}\alpha^H_{\pi_1(i)}\beta^H_{\pi_2(i)}\right)+\frac{1}{2}\max_{\pi_1,\pi_2}\left(\sum_{i=1}^{k}\alpha^S_{\pi_1(i)}\beta^S_{\pi_2(i)}\right).
\end{aligned}$$

4.4. Case of two normal matrices. In the case of normal matrices A and B (i.e., $AA^* = A^*A$ and $BB^* = B^*B$), the optimal value for the objective function can be found for $k = 1$ and $k = m = n$. For general $k \le \min(m, n)$, only an upper bound for the optimal value of the problem can be obtained. The following developments are based on the fact that all normal matrices are diagonalizable under unitary transformation. We can thus make the matrices A and B diagonal matrices D_A and D_B by unitary transformations, with $D_A = D_{A_R} + jD_{A_I}$ and $D_B = D_{B_R} + jD_{B_I}$, where the subscripts $_R$ and $_I$ denote, respectively, the real and imaginary parts of the matrices. In the rest of the section, α_i, $i = 1, \ldots, m$, and β_i, $i = 1 \ldots, n$, are the eigenvalues of A and B, respectively.

4.4.1. One-dimensional case.

THEOREM 4.4. [illegible] $k = 1$ [illegible] A [illegible] B [illegible]

$$\max_{\substack{u^*u=1\\ v^*v=1}} \langle u^*D_Au, v^*D_Bv\rangle_H = \max_{i,j} \Re(\alpha_i\beta_j). \qquad (4.17)$$

[illegible]. For $k = 1$, and by using the diagonalization of the normal matrices A and B, the maximization (4.1) can be expressed as follows:

$$\max_{\substack{u^*u=1\\ v^*v=1}} \langle u^*D_Au, v^*D_Bv\rangle_H. \qquad (4.18)$$

This problem is equivalent to

$$\begin{aligned}
&\max \Re\left(\sum_{i=1}^{n}\mu_i\alpha_i\right)\left(\sum_{i=1}^{m}\nu_i\beta_i\right)\\
&\text{s.t.}\quad \sum_i \mu_i = 1,\\
&\qquad \sum_i \nu_i = 1,\\
&\qquad \mu_i^i \ge 0,\ \nu_i \ge 0,
\end{aligned} \qquad (4.19)$$

where $\mu_i = |u_i|^2$ and $\nu_i = |v_i|^2$ are nonnegative real numbers. This amounts to optimizing the real part of the products of convex combinations of the eigenvalues of A and B. This problem is a bilinear form with respect to μ_i and ν_i. If we fix μ_i, the problem is linear in ν_i and amounts to a linear programming problem. The feasible set forms a polyhedron, and the optimal solution is situated on a vertex of this polyhedron (or on a face of the polyhedron). We then apply the same reasoning for μ_i to obtain the optimal solution. The problem is then equivalent to finding the indices i and j maximizing

$$\max_{i,j} \Re(\alpha_i\beta_j) = \max_{i,j} (\alpha_{i_R}\beta_{j_R} + \alpha_{i_I}\beta_{j_I}), \qquad (4.20)$$

where the subscripts $_R$ and $_I$ denote, respectively, the real and imaginary parts. ∎

This problem can be solved in $O(mn)$ operations by merely trying out all products.

4.4.2. Square matrices.

THEOREM 4.5. [illegible] $k = m = n$ [illegible] A [illegible] B [illegible]

$$\max_{Q^*Q=I} \langle Q^*AQ, B\rangle_H = \max_{\pi_2} \sum_{i=1}^{n} (\alpha_{i_R}\beta_{\pi_2(i)_R} + \alpha_{i_I}\beta_{\pi_2(i)_I}), \tag{4.21}$$

[illegible] $\pi_2(.)$ [illegible] $1, 2, \ldots, n$ [illegible]. By using again the diagonalization of the matrices, the maximization can be expressed as follows:

$$\max_{Q^*Q=I} \left(\langle Q^*D_{A_R}Q, D_{B_R}\rangle_H + \langle Q^*D_{A_I}Q, D_{B_I}\rangle_H\right). \tag{4.22}$$

If we develop the first term in the function and we define $d_{A_R} = \mathrm{diag}(D_{A_R})$, $d_{B_R} = \mathrm{diag}(D_{B_R})$, α_{i_R} and β_{i_R} the elements i of d_{A_R} and d_{B_R}, and q_i the row i of Q, we obtain

$$\langle Q^*D_{A_R}Q, D_{B_R}\rangle_H = \sum_{i=1}^{n} \langle |q_i|^2, d_{A_R}\rangle \beta_{i_R}.$$

The last expression is equivalent to

$$\langle \hat{Q} d_{A_R}, d_{B_R}\rangle_H,$$

where $\hat{Q}_{ij} = |Q_{ij}|^2$. $\hat{Q}$ is an orthostochastic[1] matrix and hence a type of doubly stochastic matrix, i.e., $\hat{Q}_{ij} \geq 0$ for all i, j and $\hat{Q}e = \hat{Q}^T e = e$, with e the vector whose entries are all equal to 1. The fact that the row and column sums are all $+1$ follows from the fact that the rows and columns of Q are all Euclidean unit vectors. From Birkhoff's theorem (see [11]), $\hat{Q}$ is a convex combination of permutation matrices, i.e., $\hat{Q} = \sum_{i=1}^{n!} c_i P_i$, with $\sum_{i=1}^{n!} c_i = 1$ and $c_i > 0$. The above quantity $\langle \hat{Q} d_{A_R}, d_{B_R}\rangle$ is real, and then the problem (4.22) is bounded by the maximum of

$$\langle \hat{Q} d_{A_R}, d_{B_R}\rangle + \langle \hat{Q} d_{A_I}, d_{B_I}\rangle \tag{4.23}$$

for all $\hat{Q}$ doubly stochastic matrices. We are optimizing over the set of doubly stochastic matrices, but the solution is a permutation matrix and hence corresponds to a permutation matrix Q as well. The maximal value of (4.23) is the solution of a corresponding linear programming problem. This value is simply

$$\max_{\pi_2} \sum_{i=1}^{n} (\alpha_{i_R}\beta_{\pi_2(i)_R} + \alpha_{i_I}\beta_{\pi_2(i)_I}), \tag{4.24}$$

where $\pi_2(.)$ is a permutation of $1, 2, \ldots, n$. □

In the case of Hermitian matrix A or B, the problem simplifies further and is equivalent to optimizing

$$\langle \hat{Q} d_{A_R}, d_{B_R}\rangle$$

because the eigenvalues of a Hermitian matrix are real. We retrieve then the original problem developed in section 4.2.

[1] A square matrix X of the form $X = U \circ \bar{U}$ (i.e., X is the Hadamard product of U with itself, $X_{ij} = U_{ij}^2$) for some unitary U is said to be orthostochastic.

TABLE 4.1
Summary of the results and bounds obtained for particular matrices and dimensions.

Dimensions	Matrices	Optimum/Upper bound
$k = m = n$	A Hermitian, B arbitrary	Optimum: maximal combination of the eigenvalues of A and $(B + B^*)$ (4.7): $\frac{1}{2}\max_{\pi_1,\pi_2}(\sum_{i=1}^{k}\alpha_{\pi_1(i)}\beta_{\pi_2(i)})$
	A, B normal	Optimum: maximal combination of the eigenvalues of A and B (4.24): $\max_{\pi_2}\sum_{i=1}^{n}(\alpha_{i_R}\beta_{\pi_2(i)_R} + \alpha_{i_I}\beta_{\pi_2(i)_I})$
	A, B arbitrary	Bound: solution of the semidefinite programming relaxation ν_D (3.6) or (4.15).
$k = 1$	A, B real	Optimum: product of the adequate extremal eigenvalues of the symmetric parts of A and B (section 3.2.2)
	A, B Hermitian	Optimum: product of the adequate eigenvalues of A and B (section 3.2.1)
	A, B normal	Optimum: maximal combination of the real and imaginary parts of an eigenvalue of A and an eigenvalue of B (4.20): $\max_{i,j}(\alpha_{R_i}\beta_{R_j} + \alpha_{I_i}\beta_{I_j})$
	A, B arbitrary	Bound: maximal sum of the products of the adequate extremal eigenvalues of the Hermitian and skew-Hermitian parts of A and B (4.15)
$k \leq \min(m, n)$	A Hermitian, B arbitrary	Optimum: maximal sum of the products of k eigenvalues of A and $(B + B^*)$ (4.7): $\frac{1}{2}\max_{\pi_1,\pi_2}(\sum_{i=1}^{k}\alpha_{\pi_1(i)}\beta_{\pi_2(i)})$
	A, B normal	Bound: maximal sum of the combinations of the real and imaginary parts of k eigenvalues of A and B (4.25): $\max_{\pi_1,\pi_2}\sum_{i=1}^{k}(\alpha_{\pi_1(i)_R}\beta_{\pi_2(i)_R}) +$ $\max_{\pi_1,\pi_2}\sum_{i=1}^{k}(\alpha_{\pi_1(i)_I}\beta_{\pi_2(i)_I})$
	A, B arbitrary	Bound: maximal sum of the combinations of the eigenvalues of k Hermitian and skew-Hermitian parts of A and B (4.15): $\frac{1}{2}\max_{\pi_1,\pi_2}(\sum_{i=1}^{k}\alpha^H_{\pi_1(i)}\beta^H_{\pi_2(i)}) +$ $\frac{1}{2}\max_{\pi_1,\pi_2}(\sum_{i=1}^{k}\alpha^S_{\pi_1(i)}\beta^S_{\pi_2(i)})$

4.4.3. General case. In the general case of normal matrices $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$, for $1 \leq k \leq \min(m, n)$ an upper bound for the general problem (4.1) can be found. We optimize the function

$$\max_{\substack{U^*U=I_k \\ V^*V=I_k}} \left(\langle U^*D_{A_R}U, V^*D_{B_R}V\rangle_H + \langle U^*D_{A_I}U, V^*D_{B_I}V\rangle_H\right).$$

An upper bound to this problem is then

$$\max_{\pi_1,\pi_2}\sum_{i=1}^{k}(\alpha_{\pi_1(i)_R}\beta_{\pi_2(i)_R}) + \max_{\pi_1,\pi_2}\sum_{i=1}^{k}(\alpha_{\pi_1(i)_I}\beta_{\pi_2(i)_I}), \tag{4.25}$$

where α_{i_R} and β_{i_R} are the elements of $\mathrm{diag}(D_{A_R})$ and $\mathrm{diag}(D_{B_R})$, respectively, and α_{i_I} and β_{i_I} the elements of $\mathrm{diag}(D_{A_I})$ and $\mathrm{diag}(D_{B_I})$, respectively. This problem is combinatorial and differs from (4.24).

4.5. Summary of optimal values and bounds. Table 4.1 summarizes the results and the bounds for the problem (4.1) developed in the previous sections. These results and bounds depend on the kind of matrices and their sizes.

5. Numerical computation. In this section we present first an iterative algorithm to find a critical point of (4.1). We then show the equivalence between the fixed

points of the iteration and the critical points of (4.1). At the end of the section, we present some numerical experiments of the algorithm applied to nilpotent matrices.

5.1. Algorithm. The proposed algorithm to solve (4.1) is based on the following relations:

$$\begin{aligned} F(U,V) &= \langle U^*AU, V^*BV\rangle_H \\ &= \frac{1}{2}\langle UV^*, AUV^*B^* + A^*UV^*B\rangle_H \\ &= \frac{1}{2}\left(\langle UV^*, AUV^*B^* + A^*UV^*B + sUV^*\rangle_H - sk\right) \end{aligned}$$

resulting from the properties (2.3) and where s is a constant scalar. Let us now define the linear map $M_s(X) = AXB^* + A^*XB + sX$, and then the problem of maximizing $F(U,V)$ is equivalent to the following constrained maximization problem:

$$\max_X G(X) = \langle X, M_s(X)\rangle_H \qquad \text{s.t.} \qquad X = UV^*,\ U^*U = V^*V = I_k. \tag{5.1}$$

An algorithm for this problem is given in this section, but it relies on a few intermediate results. In order to show the uniqueness of the iterates of our algorithm, we will need the following two lemmas.

LEMMA 5.1. [illegible] $M \in \mathbb{C}^{m\times n}$ [illegible]

$$M = \left[\begin{array}{c|c} P_1 & P_2 \end{array}\right] \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right] \left[\begin{array}{c|c} Q_1 & Q_2 \end{array}\right]^*,$$

[illegible] $P_1 \in \mathbb{C}^{m\times k}$, $P_2 \in \mathbb{C}^{m\times(m-k)}$, $Q_1 \in \mathbb{C}^{n\times k}$, $Q_2 \in \mathbb{C}^{n\times(n-k)}$, $\Sigma_1 \in \mathbb{R}^{k\times k}$, $\Sigma_2 \in \mathbb{R}^{(m-k)\times(n-k)}$ [illegible] $k \le \min(m,n)$ [illegible] $P_1Q_1^*$ [illegible] $\sigma_{min}(\Sigma_1) > \sigma_{max}(\Sigma_2)$, $k < \min(m,n)$ [illegible] $\sigma_{min}(\Sigma_1) > 0$, $k = \min(m,n)$ [illegible]. This is a well-known result, discussed, e.g., in [12, Theorems 3.1.1 and 3.1.1']. □

Notice that the pairs of matrices (P_1, Q_1) are not unique but are all given by (P_1R, Q_1R), where R is a unitary matrix commuting with Σ_1. But the degree of freedom R disappears in the product $P_1Q_1^*$.

LEMMA 5.2. [illegible] $a_i, b_i \in \mathbb{R}$, $i = 1,\dots,m$, [illegible] $b_1 \ge b_2 \ge \cdots \ge b_m \ge 0$ [illegible] $\sum_{i=1}^m a_i \le k$, $k \le m$, [illegible] $0 \le a_i \le 1$, $i = 1,\dots,m$, [illegible]

$$\sum_{i=1}^m a_ib_i \le \sum_{i=1}^k b_i.$$

[illegible] $a_i = 1$, $i \le k$, [illegible] $a_i = 0$, $i > k$ [illegible] [illegible] $b_k > b_{k+1}$ $(k < m)$, $b_k > 0$ $(k = m)$ [illegible]. The inequality results from Proposition B.7 for majorized sequences in [17] by remarking that $\sum_{i=1}^m a_ib_i = b^Ta$ and $\sum_{i=1}^k b_i = b^Tv$, with

$$v = (\underbrace{1,\dots,1}_{k},\underbrace{0,\dots,0}_{m-k}).$$

The upper bound is achieved if $\sum_{i=1}^k (1-a_i)b_i = \sum_{i=k+1}^m a_ib_i$. The terms are all nonnegative because $b_i \ge 0$ and $0 \le a_i \le 1$ for $i = 1,\dots,m$. The condition $\sum_{i=1}^m a_i \le k$ implies that

$$\sum_{i=1}^k a_i \le k - \epsilon, \qquad \sum_{i=k+1}^m a_i \le \epsilon,$$

with $0 \leq \epsilon$. Therefore

$$\sum_{i=1}^{k}(1-a_i)b_i \geq b_k \sum_{i=1}^{k}(1-a_i) \geq b_k\epsilon$$

and

$$\sum_{i=k+1}^{m} a_i b_i \leq b_{k+1}\epsilon.$$

If $b_k > b_{k+1}$ (or $b_k > 0$ if $k = m$), the equality $\sum_{i=1}^{k}(1-a_i)b_i = \sum_{i=k+1}^{m} a_i b_i$ is thus achieved if and only if $\epsilon = 0$, i.e., all of the terms equal 0. This happens if $a = v$, which is the unique solution. □

We now propose an algorithm to solve (5.1) by using an iteration

$$X_{i+1} = \arg\max_{X} \langle X, M_s(X_i)\rangle_H$$

which is analyzed in the following theorem.

THEOREM 5.3. [illegible] $M_s(X_i) \in \mathbb{C}^{m\times n}$ [illegible]

$$M_s(X_i) = AX_iB^* + A^*X_iB + sX_i, \qquad X_i = U_iV_i^*,\ U_i^*U_i = I_k = V_i^*V_i,$$

[illegible] $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $U_i \in \mathbb{C}^{m\times k}$, [illegible] $V_i \in \mathbb{C}^{n\times k}$, [illegible] $k \leq \min(m,n)$, [illegible] s [illegible] $s_{min} := 4\|A\|_2\|B\|_2$ [illegible] $M_s(X_i)$ [illegible]

$$M_s(X_i) = \begin{bmatrix} P_1 \mid P_2 \end{bmatrix} \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right] \begin{bmatrix} Q_1 \mid Q_2 \end{bmatrix}^* = P\Sigma Q^*, \tag{5.2}$$

[illegible] $P_1 \in \mathbb{C}^{m\times k}$, $P_2 \in \mathbb{C}^{m\times(m-k)}$, $Q_1 \in \mathbb{C}^{n\times k}$, $Q_2 \in \mathbb{C}^{n\times(n-k)}$, $\Sigma_1 \in \mathbb{R}^{k\times k}$, $\Sigma_2 \in \mathbb{R}^{(m-k)\times(n-k)}$ [illegible] $U \in \mathbb{C}^{m\times k}$, $V \in \mathbb{C}^{n\times k}$ [illegible]

$$\max_{\substack{X=UV^* \\ U^*U=I_k=V^*V}} \langle X, M_s(X_i)\rangle_H = \sum_{i=1}^{k}\sigma_i(M_s) \tag{5.3}$$

[illegible] $\{\sigma_i(M_s)\}$ [illegible] $M_s(X_i)$ [illegible] X [illegible] $P_1Q_1^*$ [illegible]. We first show that $\sigma_k(M_s) > s_{min}/2$ and $\sigma_{k+1}(M_s) \leq s_{min}/2$ and hence that there is a gap between $\sigma_k(M_s)$ and $\sigma_{k+1}(M_s)$. This is proved as follows. The k leading singular values of sX_i are given by s and the others by 0, and the largest singular value $\sigma_1(M_0)$ of $M_0(X_i) = AX_iB^* + A^*X_iB$ is upper bounded by $s_{min}/2$ because

$$\begin{aligned}\sigma_1(M_0) &= \|AX_iB^* + A^*X_iB\|_2 \\ &\leq \|AX_iB^*\|_2 + \|A^*X_iB\|_2 \\ &\leq \|A\|_2\|B\|_2 + \|A\|_2\|B\|_2 = 2\|A\|_2\|B\|_2 = s_{min}/2.\end{aligned}$$

We now apply the perturbation result (Theorem 3.3.16 in [12]) to $M_S(X_i) = M_0(X_i) + sX_i$ and obtain

$$\sigma_{k+1}(M_s) \leq \sigma_{k+1}(sX_i) + \sigma_1(M_0) \leq s_{min}/2$$

and

$$\sigma_k(M_s) \geq \sigma_k(sX_i) - \sigma_1(M_0) > s_{min}/2.$$

We have

$$\langle X, M_s(X_i)\rangle_H = \langle X, P\Sigma Q^*\rangle_H = \langle P^*XQ, \Sigma\rangle_H,$$

and the following two problems are equivalent:

$$\max_{\substack{X=UV^* \\ U^*U=I_k=V^*V}} \langle X, M_s(X_i)\rangle_H = \max_{\substack{\tilde{X}=\tilde{U}\tilde{V}^* \\ \tilde{U}^*\tilde{U}=I_k=\tilde{V}^*\tilde{V}}} \langle \tilde{X}, \Sigma\rangle_H,$$

where $\tilde{U} = P^*U$ and $\tilde{V} = Q^*V$. Without loss of generality, we assume that $n \leq m$ (otherwise the proof is very similar). Then

$$\langle \tilde{X}, \Sigma\rangle_H = \sum_{i=1}^{n} \Re(\tilde{X}_{ii})\sigma_i(M_s) \leq \sum_{i=1}^{n} |\tilde{X}_{ii}\sigma_i(M_s)| \leq \sum_{i=1}^{n} \sigma_i(\tilde{X}^*\Sigma) \leq \sum_{i=1}^{k} \sigma_i(M_s) \tag{5.4}$$

according to Formula 3.3.10b and Lemma 3.3.1 in [12]. Moreover, since $\tilde{U}$ and $\tilde{V}$ are isometries, we have

$$0 \leq |\tilde{X}_{ii}| \leq 1, \qquad i = 1, \ldots, n,$$

and, by a reasoning similar to (5.4) with a matrix $M_s(X_i)$ for which $\sigma_i(M_s) = 1$, $i = 1, \ldots, n$,

$$\sum_{i=1}^{n} |\tilde{X}_{ii}| \leq k.$$

According to Lemma 5.2, $\sum_{i=1}^{n} |\tilde{X}_{ii}|\sigma_i(M_s) = \sum_{i=1}^{k} \sigma_i(M_s)$ in (5.4) when $|\tilde{X}_{ii}| = 1$ for $i = 1, \ldots, k$ and $|\tilde{X}_{ii}| = 0$ for $i > k$. The upper bound in (5.4) is achieved if all inequalities are equalities. This implies that $\Re(\tilde{X}_{ii}) = |\tilde{X}_{ii}| = 1$, $i = 1, \ldots, k$, and $\Re(\tilde{X}_{ii}) = |\tilde{X}_{ii}| = 0$, $i > k$, i.e., if and only if $\tilde{X}_{ii} = 1$, $i = 1, \ldots, k$, and $\tilde{X}_{ii} = 0$, $i > k$. Since $\tilde{X} = \tilde{U}\tilde{V}^*$ and $\tilde{U}$ and $\tilde{V}$ are isometries, it happens only when $\tilde{X} = \tilde{U}\tilde{V}^* = P^*UV^*Q = \left(\begin{smallmatrix} I_k & 0 \\ 0 & 0 \end{smallmatrix}\right)$, i.e., when $X = UV^* = P_1Q_1^*$. The conditions of Lemma 5.1 are satisfied, and the solution X is therefore unique. ∎

The proposed iterative algorithm to solve (4.1) is then the following one. Choose initial isometries U_0, V_0, and, for $i \geq 0$ until convergence, compute

$$X_{i+1} = f_s(X_i) = \arg \max_{\substack{X=UV^* \\ U^*U=I_k=V^*V}} \langle X, M_s(X_i)\rangle_H, \tag{5.5}$$

where we assume that $\sigma_k(M_s) > \sigma_{k+1}(M_s)$, which is always satisfied by choosing adequately s. Theorem 5.3 gives the maximizing solution and shows that it is unique. In practice, we apply the following procedure in which we switch again to the formulation in terms of U and V: Choose initial isometries U_0 and V_0 and a value for s, and, for $i = 0, 1, \ldots$ until convergence, compute the SVD:

$$\left[\begin{array}{c|c} U_{i+1} & U_\perp \end{array}\right] \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right] \left[\begin{array}{c|c} V_{i+1} & V_\perp \end{array}\right]^* = AU_iV_i^*B^* + A^*U_iV_i^*B + sU_iV_i^*.$$

When the product $U_iV_i^*$ converges (i.e., when $\|U_iV_i^* - U_{i+1}V_{i+1}^*\| \to 0$), there exists a diagonal matrix $\Lambda = \Sigma_1 - sI_k$ such that

$$U\Lambda V^* = AUV^*B^* + A^*UV^*B - U_\perp \Sigma_2 V_\perp^*, \tag{5.6}$$

where $U_\perp$ and $V_\perp$ are matrices such that every column of $U_\perp$ (resp., $V_\perp$) is orthogonal to every column of U (resp., V). Σ_2 is a diagonal matrix with elements which are all smaller than the elements of Σ_1. The scalar s must be larger than $s_{min} = 4\|A\|_2\|B\|_2$. The convergence is not proved, but in all experiments the process always converged linearly to a solution.

[illegible] 1. An indication that the method has typically linear convergence can be seen from the case $k = 1$ and one of the matrices Hermitian (say, $A = A^*$) (with $|\alpha_1| \geq |\alpha_i|, i = 1, \ldots, m$, $|\beta_1| \geq |\beta_i|, i = 1, \ldots, n$, and $\alpha_1\beta_1 > 0$ to simplify the reasoning; indeed the parameter s could then be zero). In this particular case, the iteration becomes

$$u_{i+1}\sigma_{i+1}v_{i+1}^* = Au_iv_i^*(B + B^*) \tag{5.7}$$

because the right-hand side is exactly of rank one. This algorithm corresponds to the combination of two power methods for A and $B + B^*$. Linear convergence is thus guaranteed. The reasoning could be extended for arbitrary $k \leq \min(m, n)$ and for $s = 0$ in the iteration.

5.2. Relation to the optimization problem. In this part we show that solving the iteration (5.5) is equivalent to solving the optimization problem (4.1), whose critical points are expressed by (4.5).

THEOREM 5.4. [illegible] $s \geq s_{min}$, [illegible] $M_s(UV^*)$ [illegible] $f_s(UV^*)$ [illegible] (U, V) [illegible] $F(U, V)$ [illegible] (U, V) [illegible] $F(U, V)$ [illegible] UV^* [illegible] $f_s(UV^*)$

[illegible]. Let UV^* be a fixed point of $f_s(UV^*)$. Then according to Theorem 5.3

$$U\Sigma_1V^* + U_\perp\Sigma_2V_\perp^* = AUV^*B^* + A^*UV^*B + sUV^*.$$

Multiply this matrix by V and its Hermitian conjugate by U to get

$$U(\Sigma_1 - sI_k) = AUV^*B^*V + A^*UV^*BV,$$
$$V(\Sigma_1 - sI_k) = BVU^*A^*U + B^*VU^*AU.$$

This is nothing but the condition for a critical point (U, V) of $F(U, V)$ (see (4.5)).

Conversely, let (U, V) be a critical point of $F(U, V)$, and then

$$U\Lambda = AU(V^*B^*V) + A^*U(V^*BV),$$
$$V\Lambda = BV(U^*A^*U) + B^*V(U^*AU),$$

where we have chosen the diagonal ordered form for Λ (see (4.5)). Then, if we define $\Sigma_1 = \Lambda + sI$, with $s > 4\|A\|_2\|B\|_2$,

$$U\Sigma_1 = M_s(UV^*)V, \qquad \Sigma_1V^* = U^*M_s(UV^*).$$

These equations express that the diagonal elements of Σ_1 are singular values of $M_s(UV^*)$ and that the columns of U and V are corresponding right and left singular vectors. Hence there exists an SVD

$$M_s(UV^*) = \left[\begin{array}{c|c} U & U_\perp \end{array}\right] \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right] \left[\begin{array}{c|c} V & V_\perp \end{array}\right]^*,$$

TABLE 5.1
Conjectured values for (5.10) *given in* [9].

n	1	2	3	4	5	6
f_n^{max}	2	4	$4(1+\sqrt{3})$	$8(1+\sqrt{3})$	$16(1+\sqrt{3})+4\sqrt{5}$	$32(1+\sqrt{3})+8\sqrt{5}$

where $\Sigma_2, U_\perp, V_\perp$ contain the remaining singular values and vectors of $M_s(UV^*)$ (where we can choose to order the diagonal elements of Σ_2 as well). Moreover $\sigma_{min}(\Sigma_1) > 2\|A\|_2\|B\|_2 \geq \sigma_{max}(\Sigma_2)$ (see proof of Theorem 5.3). This is the condition for a fixed point UV^* of $f_s(UV^*)$, according to Theorem 5.3. □

Remark that, for arbitrary matrices A and B, local minima and local maxima can exist, and then the algorithm may not always converge to the global optimum of the function.

5.3. Numerical experiments. As an illustration of the algorithm, we consider the problem of maximizing (4.1) in the case of special nilpotent matrices A_n and B_n. For any $n \in \mathbb{N}$, the nilpotent $(2^{n+1} \times 2^{n+1})$ matrices are recursively defined by

$$A_n = \begin{pmatrix} N_n & 0 \\ 0 & N_n \end{pmatrix}, \qquad B_n = \begin{pmatrix} 0 & 0 \\ I_{2^n} & 0 \end{pmatrix}, \tag{5.8}$$

with I_m the $m \times m$ identity matrix and N_n given inductively by

$$N_n = \begin{pmatrix} N_{n-1} & 0 \\ I_{2^{n-1}} & N_{n-1} \end{pmatrix}, \qquad N_0 = 0. \tag{5.9}$$

The matrices U and V are chosen of the same dimension $2^{n+1} \times 2^{n+1}$. The problem (3.1) is equivalent to maximizing the C-numerical range of A_n

$$\max_{Q^*Q=I} \langle Q^* A_n Q, B_n \rangle_H. \tag{5.10}$$

In [9], the authors provide conjectured maximal values of the function, depending on n. These values are represented in Table 5.1 and have been proved to be correct for $n = 1, 2$.

5.3.1. Numerical values by application of the algorithm. We apply the algorithm given in section 5.1 for nilpotent matrices A_n and B_n defined above (for $n = 1, \ldots, 6$). Initial unitary matrices U_0 and V_0 are randomly generated. The results are presented in Figure 5.1. Each plot combines the trajectories for three different initial values. The function

$$\text{residual} = f_n^{max} - \max_{\substack{U^*U=I \\ V^*V=I}} \langle U^* A_n U, V^* B_n V \rangle_H \tag{5.11}$$

is plotted on a logarithmic scale against the number of iterations. The values of f_n^{max} are taken from the above conjecture. The termination criteria we used for the different plots are represented in Table 5.2. We observe a convergence to a maximum defined by the values of the conjecture given in [9].

5.3.2. Duality gap. In this part we show that a duality gap can occur for the problem (3.6) in the case of nonsymmetric matrices. Consider (3.6) for nilpotent matrices A_3 and B_3. In the assumption that the conjecture in [9] is true, $\nu_P = 4(1+\sqrt{3}) = 10.92$. The value obtained for the dual problem is $\nu_D = 11$. That proves that a nonduality gap can occur, i.e., $\nu_P \leq \nu_D$.

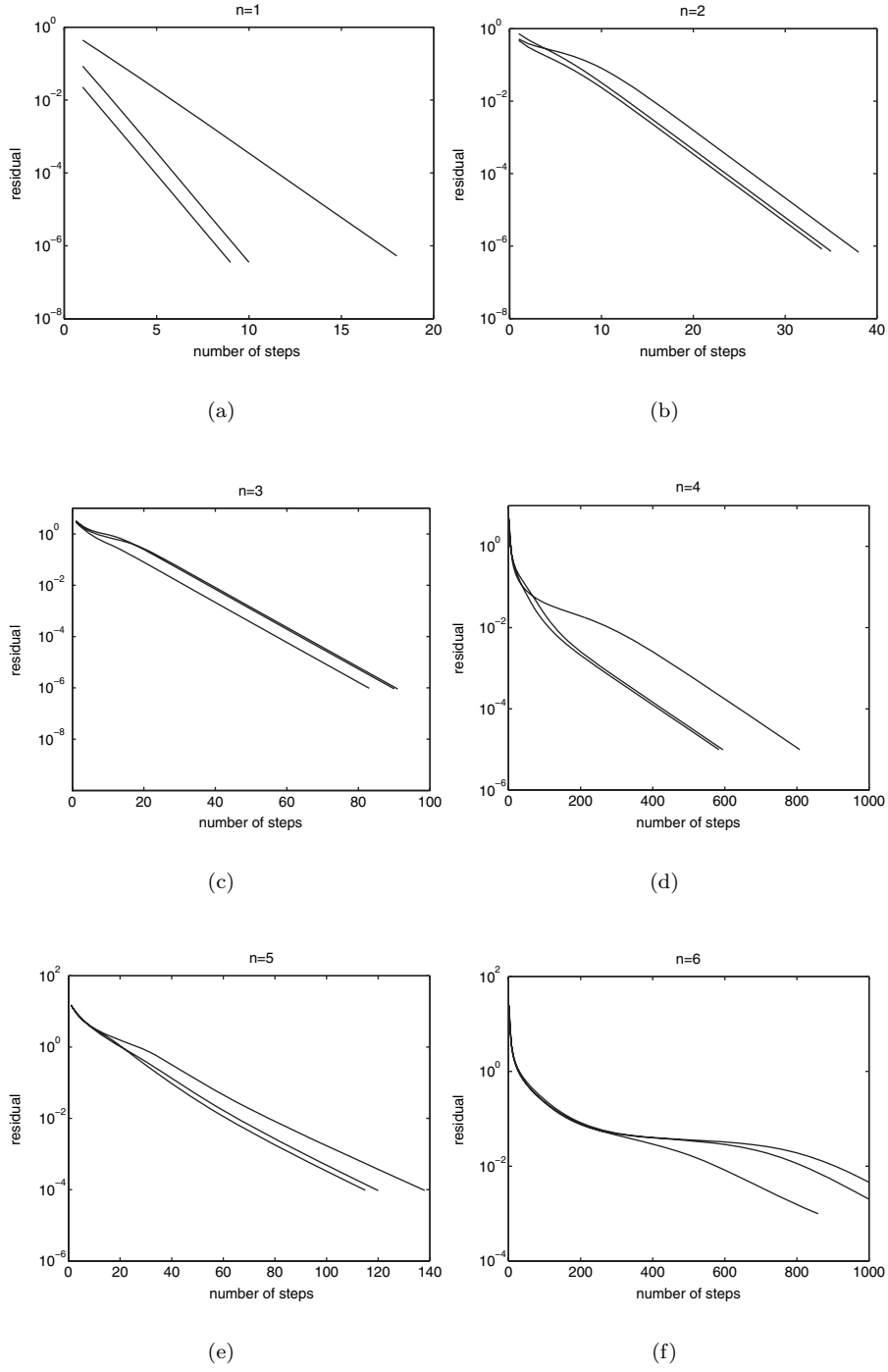


FIG. 5.1. *Minimization of the residual* (5.11) *by application of the SVD algorithm.*

TABLE 5.2
Termination criteria for the plots.

n	Residual	or	Number of steps
1	$< 10^{-6}$		> 1000
2	$< 10^{-6}$		> 1000
3	$< 10^{-6}$		> 1000
4	$< 10^{-5}$		> 1000
5	$< 10^{-4}$		> 1000
6	$< 10^{-3}$		> 1000

6. Conclusion. In this paper, we analyze the coupling between two restricted matrices under isometry constraints. Our problem provides a method to project simultaneously the matrices in a subspace of arbitrary dimension k and can be applied to both real and complex matrices. We indicate that it is an extension of various problems found in the literature. Many applications can arise from this formulation.

We present some mathematical properties of the problem and we characterize the maximal coupling for particular matrices such as Hermitian or normal matrices. In general only an upper bound can be found theoretically.

We develop an iterative algorithm in order to reach the optimum, and we characterize the fixed points. This algorithm is very simple to implement and is based on the singular value decomposition. Because this problem is not convex, the analysis of convergence and stability of the fixed points is difficult to realize.

Investigations of mathematical properties and applications of the similarity between restricted matrices can be pursued in several directions. A deeper analysis of the convergence of the algorithm is worthwhile to consider. We outline in the rest of the section a nonexhaustive list of some possible improvements and future research directions.

The first possible improvement concerns the convergence of the algorithm. Experimentally we observe a linear convergence to the optimum, but this convergence has not yet been proved and remains an important point to develop in the future. Second, because the problem is not convex, the analysis of the stability of the fixed points and the study of their basins of attraction are not easy to obtain. This last point is thus a delicate but interesting task to explore. From a more applied point of view, another topic of interest is to investigate how the mathematical concepts proposed here can be used, possibly in modified form, for applications in various areas. Some research for the use of the algorithm in the graph matching problem has been initiated but still needs further investigation. We can conclude that the problem envisaged in this paper gives rise to the study of interesting mathematical properties but also to various applications in different areas.

Acknowledgment. The authors thank the anonymous referee whose enormous work largely improved this paper.

REFERENCES

[1] K. ANSTREICHER AND H. WOLKOWICZ, *On Lagrangian relaxation of quadratic matrix constraints*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 41–55.

[2] V. D. BLONDEL, A. GAJARDO, M. HEYMANS, P. SENELLART, AND P. VAN DOOREN, *A measure of similarity between graph vertices: Applications to synonym extraction and Web searching*, SIAM Rev., 46 (2004), pp. 647–666.

[3] R. W. Brockett, *Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems*, Linear Algebra Appl., 146 (1991), pp. 79–91.
[4] T. Caelli and S. Kosinov, *An eigenspace projection clustering method for inexact graph matching*, IEEE Trans. Pattern Analysis and Machine Intelligence, 26 (2004), pp. 515–519.
[5] W.-S. Cheung and N. K. Tsing, *The C-numerical range of matrices is star-shaped*, Linear Multilinear Algebra, 41 (1996), pp. 245–250.
[6] D. Conte, P. Foggia, C. Sansone, and M. Vento, *Thirty years of graph matching in pattern recognition*, Int. J. Pattern Recognition and Artificial Intelligence, 18 (2004), pp. 265–298.
[7] S. J. Glaser, T. Schulte-Herbrüggen, M. Sieveking, O. Schedletzky, N. C. Nielsen, O. W. Sørensen, and C. Griesinger, *Unitary control in quantum ensembles: Maximizing signal intensity in coherent spectroscopy*, Science, 280 (1998), pp. 421–424.
[8] M. Goldberg and E. G. Straus, *Elementary inclusion relations for generalized numerical ranges*, Linear Algebra Appl., 18 (1977), pp. 1–24.
[9] U. Helmke, K. Hüper, J. B. Moore, and Th. Schulte-Herbrüggen, *Gradient flows computing the C-numerical range with applications in NMR spectroscopy*, J. Global Optim., 23 (2002), pp. 283–308.
[10] U. Helmke and J. Moore, *Optimization and Dynamical Systems*, Springer-Verlag, Berlin, 1994.
[11] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.
[12] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.
[13] C. K. Li, *C-numerical ranges and C-numerical radii*, Linear Multilinear Algebra, 37 (1994), pp. 51–82.
[14] C. K. Li and L. Rodman, *Multiplicative preservers of C-numerical ranges and radii*, Linear Multilinear Algebra, 52 (2004), pp. 265–279.
[15] C. K. Li and N. K. Tsing, *Norms that are invariant under unitary similarities and the C-numerical radii*, Linear Multilinear Algebra, 24 (1989), pp. 209–222.
[16] C.-K. Li and H. J. Woerdeman, *A lower bound on the C-numerical radius of nilpotent matrices appearing in coherent spectroscopy*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 793–800.
[17] A. Marshall and I. Olkin, *Inequalities: Theory of Majorization and its Applications*, Academic Press, New York, 1979.
[18] F. Rendl and H. Wolkowicz, *Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem*, Math. Program., 53 (1992), pp. 63–78.
[19] R. Westwick, *A theorem on numerical range*, Linear Multilinear Algebra, 2 (1975), pp. 311–315.
[20] Q. Zhao, S. E. Karisch, F. Rendl, and H. Wolkowicz, *Semidefinite programming relaxations for the quadratic assignment problem*, J. Comb. Optim., 2 (1998), pp. 71–109.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 346–363

CONVERGENCE OF THE DOMINANT POLE ALGORITHM AND RAYLEIGH QUOTIENT ITERATION*

JOOST ROMMES† AND GERARD L. G. SLEIJPEN‡

Abstract. The dominant poles of a transfer function are specific eigenvalues of the state space matrix of the corresponding dynamical system. In this paper, two methods for the computation of the dominant poles of a large scale transfer function are studied: two-sided Rayleigh quotient iteration (RQI) and the dominant pole algorithm (DPA). First, a local convergence analysis of DPA will be given, and the local convergence neighborhoods of the dominant poles will be characterized for both methods. Second, theoretical and numerical results will be presented that indicate that for DPA the basins of attraction of the dominant pole are larger than those for two-sided RQI. The price for the better global convergence is only a few additional iterations, due to the asymptotically quadratic rate of convergence of DPA, against the cubic rate of two-sided RQI.

Key words. eigenvalues, eigenvectors, dominant poles, two-sided Rayleigh quotient iteration, dominant pole algorithm, subspace accelerated Newton method, rate of convergence, transfer function, modal model reduction

AMS subject classification. 65F15

DOI. 10.1137/060671401

1. Introduction.

The transfer function of a large scale dynamical system often has only a small number of dominant poles compared to the number of state variables. The computation of the dominant poles that are specific eigenvalues of the system matrix, and the corresponding modes, requires specialized eigenvalue methods. In [15] Newton's method is used to compute a dominant pole of single input single output (SISO) transfer function: the dominant pole algorithm (DPA). In two recent publications this algorithm is improved and extended to a robust and efficient method for the computation of the dominant poles and modes of large scale SISO [23] and MIMO [22] transfer functions.

This paper is concerned with the convergence behavior of DPA. First, DPA will be related to the two-sided or generalized Rayleigh quotient iteration (RQI) [18, 20]. A local convergence analysis will be given, showing the asymptotically quadratic rate of convergence. Furthermore, for systems with a symmetric state-space matrix, a characterization of the local convergence neighborhood of the dominant pole will be presented for both DPA and RQI. The results presented in this paper are sharp (in some sense), in contrast to those found in the literature, for example, by Ostrowski [17, 18] for DPA, and by Beattie and Fox [5] for RQI. Second, theoretical and numerical results indicate that for DPA the basins of attraction of the most dominant poles are larger than for two-sided RQI. In practice, the asymptotically quadratic (DPA) instead of cubic rate (two-sided RQI) of convergence costs about two or three iterations.

The outline of this paper is as follows. Definitions and properties of transfer functions and dominant poles and further motivation are given in section 2. The

*Received by the editors October 4, 2006; accepted for publication (in revised form) by A. Frommer November 12, 2007; published electronically March 19, 2008. This work was supported by the BRICKS-MSV1 project.

http://www.siam.org/journals/simax/30-1/67140.html

†NXP Semiconductors, Corp. I&T/DTF, HTC 37 WY4-01, 5656 AE, Eindhoven, The Netherlands (joost.rommes@nxp.com). This work was done at the Mathematical Institute, Utrecht University.

‡Mathematical Institute, Utrecht University, P.O. Box 80010, 3508 TA, Utrecht, The Netherlands (sleijpen@math.uu.nl, http://www.math.uu.nl/people/sleijpen).

dominant pole algorithm (DPA) and its relation to the two-sided Rayleigh quotient iteration (RQI) are discussed in section 3. In section 4 the local convergence of DPA is analyzed. The basins of attraction of DPA and two-sided RQI are studied in section 5. Section 6 concludes the paper.

2. Transfer functions and poles. The motivation for this paper comes from dynamical systems $(A, E, \mathbf{b}, \mathbf{c}, d)$ of the form

$$\left\{ \begin{array}{rcl} E\dot{\mathbf{x}}(t) & = & A\mathbf{x}(t) + \mathbf{b}u(t), \\ y(t) & = & \mathbf{c}^*\mathbf{x}(t) + du(t), \end{array} \right. \tag{2.1}$$

where $A, E \in \mathbb{R}^{n\times n}$, E may be singular but the pencil (A, E) is regular, $\mathbf{b}, \mathbf{c}, \mathbf{x}(t) \in \mathbb{R}^n$, and $u(t), y(t), d \in \mathbb{R}$. The vectors $\mathbf{b}$ and $\mathbf{c}$ are called the input and output vectors, respectively. The transfer function $H : \mathbb{C} \longrightarrow \mathbb{C}$ of (2.1) is defined as

$$H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b} + d. \tag{2.2}$$

The poles of transfer function (2.2) are a subset of the eigenvalues $\lambda_i \in \mathbb{C}$ of the matrix pencil (A, E). An eigentriplet $(\lambda_i, \mathbf{v}_i, \mathbf{w}_i)$ is composed of an eigenvalue λ_i of (A, E) and corresponding right and left eigenvectors $\mathbf{v}_i, \mathbf{w}_i \in \mathbb{C}^n$:

$$\begin{aligned} A\mathbf{v}_i &= \lambda_i E\mathbf{v}_i, \qquad \mathbf{v}_i \neq 0, \\ \mathbf{w}_i^*A &= \lambda_i \mathbf{w}_i^*E, \qquad \mathbf{w}_i \neq 0. \end{aligned}$$

It is well known that left and right eigenvectors corresponding to distinct eigenvalues are E-orthogonal: $\mathbf{w}_i^*E\mathbf{v}_j = 0$ if $\lambda_i \neq \lambda_j$.

If the pencil is nondefective, then the eigentriplets may be selected such that the first $\tilde{n}$ eigenvalues are distinct and finite ($\lambda_i \neq \lambda_j$ if $i \neq j$, $i, j \leq \tilde{n}$):

$$\mathbf{b} = \sum_{i=1}^{\tilde{n}} \beta_i E\mathbf{v}_i + \beta_\infty A\mathbf{v}_\infty, \quad \text{and} \quad \mathbf{c} = \sum_{i=1}^{\tilde{n}} \gamma_i E^*\mathbf{w}_i + \gamma_\infty A^*\mathbf{w}_\infty;$$

i.e., $\mathbf{b}$ and $\mathbf{c}$ determine the eigenvectors to be selected in the eigenspaces $\{\mathbf{v} \mid A\mathbf{v} = \lambda_i E\mathbf{v}\}$ and $\{\mathbf{w} \mid \mathbf{w}^*A = \lambda_i \mathbf{w}^*E\}$, respectively. The vectors $\mathbf{v}_\infty$ and $\mathbf{w}_\infty$ are in the kernel of E and E^*, respectively. They correspond to the eigenvalue at ∞. It is assumed that both $\mathbf{v}_i$ and $\mathbf{w}_i$ are nonzero and at least one of the coefficients β_i or γ_i is nonzero. The right and left eigenvectors $\mathbf{v}_i$ and $\mathbf{w}_i$ with $\mathbf{w}_i^*E\mathbf{v}_i \neq 0$ corresponding to finite eigenvales are assumed to be scaled so that $\mathbf{w}_i^*E\mathbf{v}_i = 1$.

The transfer function $H(s)$ can be expressed as a sum of residues $R_i \in \mathbb{C}$ [14]:

$$H(s) = \sum_{i=1}^{\tilde{n}} \frac{R_i}{s - \lambda_i} + R_\infty + d, \tag{2.3}$$

where the residues R_i are

$$R_i = (\mathbf{c}^*\mathbf{v}_i)(\mathbf{w}_i^*\mathbf{b}),$$

R_∞ (which is often zero) is the constant contribution of the poles at infinity, and $\tilde{n} \leq n$ is the number of finite first order poles (to be assumed to be numbered first). Note that $R_i = 0$ if $\mathbf{w}_i^*E\mathbf{v}_i = 0$.

Although there are different indices of modal dominance [2, 10, 23, 29], the following [11] will be used in this paper.

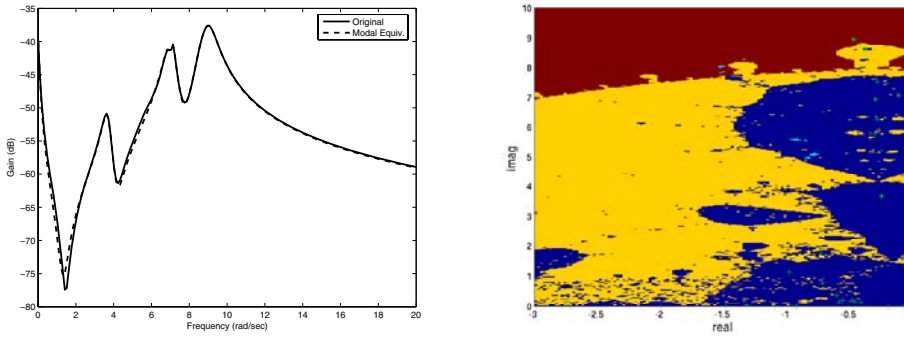


FIG. 2.1. *The left figure shows the Bode plot of the transfer function ($n = 66$ states) of the New England test system* [15]*, together with the Bode plot of the $k = 11$th order modal equivalent, constructed by projecting the system onto the modes of the 6 most dominant poles, which may belong to complex conjugated pairs. The right figure shows part of the complex plane with part of the pole spectrum (dominant poles are marked by asterisks, other poles by plus-signs), together with the initial shifts for which DPA (marked yellow and red grid points) and two-sided RQI (light-blue and red) converge to the most dominant pole $\lambda = -0.467 \pm 8.96i$. Dark blue regions denote convergence to less dominant poles. Real (imaginary) parts of initial shifts are at the horizontal (vertical) axis.*

DEFINITION 2.1. [illegible] λ_i [illegible] $H(s)$ [illegible] $\mathbf{v}_i$ [illegible] $\mathbf{w}_i$ ($\mathbf{w}_i^* E \mathbf{v}_i = 1$) [illegible] $|R_i| > |R_j|$ [illegible] $j \neq i$

An approximation of $H(s)$ that consists of $k < n$ terms with $|R_j|$ above some value determines the effective transfer function behavior [25] and is also known as the transfer function modal equivalent (assuming $R_\infty = 0$):

$$H_k(s) = \sum_{j=1}^{k} \frac{R_j}{s - \lambda_j} + d.$$

More generally, a pole λ_i is called dominant if $|R_i|$ is not very small compared to $|R_j|$ for all $j \neq i$. A dominant pole can be well observable and controllable in the transfer function. Its presence can be observed in the Bode plot corresponding to (2.2) (see Figure 2.1), which is a plot of $|H(i\omega)|$ against $\omega \in \mathbb{R}$: in this example, peaks occur at frequencies ω close to the imaginary parts of the dominant poles of $H(s)$. The height of the peaks, and the controllability/observability of the (dominant) pole that causes the peak, also depends on the size of the real part of that pole (cf. (2.3)). Therefore, in light of model order reduction by modal truncation, Definition 2.1 may not be suitable, and a characterization in terms of $|R_i|/|\mathrm{Re}(\lambda_i)|$ might be more appropriate. The purpose of this paper, however, is to analyze the convergence of the dominant pole algorithm (DPA) [15], described in the following section, and compare it to the convergence of the RQI. For this purpose, Definition 2.1 will do. For an overview of model order reduction techniques, see [3].

The dominant poles are specific (complex) eigenvalues of the pencil (A, E) and usually form a small subset of the spectrum of (A, E). They can be located anywhere in the spectrum; see also Figure 2.1. The two algorithms to compute poles (eigenvalues) that will be discussed in this paper, DPA and two-sided RQI, both start with an initial shift s_0 but behave notably differently: as can be seen in Figure 2.1, DPA converges to the most dominant pole for many more initial shifts than two-sided RQI (marked by yellow and red points, and light-blue and red points, respectively;

red denotes convergence for both DPA and two-sided RQI, while dark-blue denotes convergence to less dominant poles). In section 5 more of such figures will be presented and for all figures the following holds: the larger the yellow areas (compared to light-blue areas), the better the performance of DPA over two-sided RQI. The typical behavior of DPA will be discussed in more detail in sections 4 and 5.

In both DPA and two-sided RQI, the [illegible] plays a central role. This quotient is defined as follows (cf. [18, 20]).

DEFINITION 2.2. [illegible] $\rho(\mathbf{x},\mathbf{y})$ [illegible] $\rho(\mathbf{x},\mathbf{y}) \equiv \rho(\mathbf{x},\mathbf{y},A,E) \equiv \mathbf{y}^*A\mathbf{x}/\mathbf{y}^*E\mathbf{x}$, [illegible] $\mathbf{y}^*E\mathbf{x} \neq 0$

Note that $\mathbf{y}^*E\mathbf{x}$ can be zero even if E is nonsingular.

Since the dominance of a pole is independent of d (where d is as in (2.3)), without loss of generality $d = 0$ in the following.

3. The dominant pole algorithm (DPA). The poles of transfer function (2.2) are the $\lambda \in \mathbb{C}$ for which $\lim_{s\to\lambda} |H(s)| = \infty$. Consider now the function $G : \mathbb{C} \longrightarrow \mathbb{C}$:

$$G(s) = \frac{1}{H(s)}.$$

For a pole λ of $H(s)$, $\lim_{s\to\lambda} G(s) = 0$. In other words, the poles are the roots of $G(s)$, and a good candidate to find these roots is Newton's method. This idea is the basis of DPA [15] (and can be generalized to MIMO systems as well; see [16, 22]).

The derivative of $G(s)$ with respect to s is given by

$$G'(s) = -\frac{H'(s)}{H^2(s)}. \tag{3.1}$$

The derivative of $H(s)$ with respect to s is

$$H'(s) = -\mathbf{c}^*(sE - A)^{-1}E(sE - A)^{-1}\mathbf{b}. \tag{3.2}$$

Equations (3.1) and (3.2) lead to the following Newton scheme:

$$\begin{aligned} s_{k+1} &= s_k - \frac{G(s_k)}{G'(s_k)} \\ &= s_k + \frac{1}{H(s_k)}\frac{H^2(s_k)}{H'(s_k)} \\ &= s_k - \frac{\mathbf{c}^*(s_kE - A)^{-1}\mathbf{b}}{\mathbf{c}^*(s_kE - A)^{-1}E(s_kE - A)^{-1}\mathbf{b}}. \end{aligned} \tag{3.3}$$

The formula (3.3) was originally derived in [6, 7]. Using $\mathbf{x}_k = (s_kE - A)^{-1}\mathbf{b}$ and $\mathbf{y}_k = (s_kE - A)^{-*}\mathbf{c}$, the Newton update (3.3) can also be written as the generalized two-sided Rayleigh quotient $\rho(\mathbf{x}_k, \mathbf{y}_k)$, provided $\mathbf{y}_k^*E\mathbf{x}_k \neq 0$:

$$\begin{aligned} s_{k+1} &= s_k - \frac{\mathbf{c}^*(s_kE - A)^{-1}\mathbf{b}}{\mathbf{c}^*(s_kE - A)^{-1}E(s_kE - A)^{-1}\mathbf{b}} \\ &= \frac{\mathbf{c}^*(s_kE - A)^{-1}A(s_kE - A)^{-1}\mathbf{b}}{\mathbf{c}^*(s_kE - A)^{-1}E(s_kE - A)^{-1}\mathbf{b}} \\ &= \frac{\mathbf{y}_k^*A\mathbf{x}_k}{\mathbf{y}_k^*E\mathbf{x}_k}. \end{aligned}$$

Algorithm 1. The dominant pole algorithm (DPA).

INPUT: System $(A, E, \mathbf{b}, \mathbf{c})$, initial pole estimate s_0, tolerance $\epsilon \ll 1$
OUTPUT: Dominant pole λ and corresponding right and left eigenvectors $\mathbf{v}$ and $\mathbf{w}$

1. Set $k = 0$
2. **while** not converged
3. Solve $\mathbf{x}_k \in \mathbb{C}^n$ from $(s_k E - A)\mathbf{x}_k = \mathbf{b}$
4. Solve $\mathbf{y}_k \in \mathbb{C}^n$ from $(s_k E - A)^* \mathbf{y}_k = \mathbf{c}$
5. Compute the new pole estimate

$$s_{k+1} = s_k - \frac{\mathbf{c}^* \mathbf{x}_k}{\mathbf{y}_k^* E \mathbf{x}_k} = \frac{\mathbf{y}_k^* A \mathbf{x}_k}{\mathbf{y}_k^* E \mathbf{x}_k}$$

6. The pole $\lambda = s_{k+1}$ with $\mathbf{v} = \mathbf{x}_k$ and $\mathbf{w} = \mathbf{y}_k$ has converged if

$$\|A\mathbf{x}_k - s_{k+1} E \mathbf{x}_k\|_2 < \epsilon$$

7. Set $k = k + 1$
8. **end while**

An implementation of this Newton scheme is represented in Algorithm 1. It is also known as the dominant pole algorithm [15].

The two linear systems that need to be solved in steps 3 and 4 of Algorithm 1 can be efficiently solved using one LU-factorization $LU = s_k E - A$, by noting that $U^* L^* = (s_k E - A)^*$. In this paper it will be assumed that an exact LU-factorization is available, although this may not always be the case for real-life examples, depending on the size and condition of the system. If an exact LU-factorization is not available, one has to use inexact Newton schemes, such as inexact RQI and Jacobi–Davidson style methods [24, 13, 26], a topic that is described in [21].

With $\mathbf{r}_k \equiv A\mathbf{x}_k - s_{k+1} E \mathbf{x}_k$ and $\Delta A \equiv \frac{1}{\mathbf{y}_k^* E \mathbf{x}_k} \mathbf{r}_k \mathbf{y}_k^* E$, $(s_{k+1}, \mathbf{x}_k)$ is an exact (right) eigenpair of the pencil $(A - \Delta A, E)$, showing that the so-called [illegible] can be bounded by $\|\Delta A\|_2 \leq \frac{1}{|\mathbf{y}_k^* E \mathbf{x}_k|} \|\mathbf{r}_k\|_2 \, \|E^* \mathbf{y}_k\|_2$. This expression can be used to adopt the stopping criterions in Algorithm 1 (step 6) and Algorithm 2 (step 7) to accommodate for a backward error less than $\epsilon \|A\|_2$. Symmetric versions, treating $\mathbf{x}_k$ and $\mathbf{y}_k$ equally, are possible; other, more convenient, norms such as $\|\cdot\|_\infty$ can be selected, and perturbations on E can be allowed as well. For a thorough discussion, see [9] and [12].

3.1. DPA and two-sided Rayleigh quotient iteration (RQI). In Algorithm 2, the two-sided RQI [18, 20] is shown. The only difference with DPA is that the right-hand sides in steps 3 and 4 of Algorithm 1 are kept fixed, while the right-hand sides in steps 4 and 5 of Algorithm 2 are updated at every iteration.

While the use of the fixed right-hand sides drops the asymptotic convergence rate from cubic to quadratic, it is exactly this use of fixed right-hand sides that causes the typical better convergence to dominant poles, as will be shown later. In that light the quadratic instead of cubic local convergence, that in practice makes only a small difference in the number of iterations, is even more acceptable. Moreover, based on criteria in [5, 27] for switching from inverse iteration to RQI, one could define similar criteria to switch from DPA to two-sided RQI in the final phase of the process, to save some iterations. However, such techniques are not considered in this paper, since the primary goal is to study the convergence behavior.

Algorithm 2. Two-sided Rayleigh quotient iteration (RQI)

INPUT: System $(A, E, \mathbf{b}, \mathbf{c})$, initial pole estimate s_0, tolerance $\epsilon \ll 1$
OUTPUT: Pole λ and corresponding right and left eigenvectors $\mathbf{v}$ and $\mathbf{w}$

1. $\mathbf{x}_0 = (s_0E - A)^{-1}\mathbf{b}$, $\mathbf{y}_0 = (s_0E - A)^{-*}\mathbf{c}$, and $s_1 = \rho(\mathbf{x}_0, \mathbf{y}_0)$
2. Set $k = 1$
3. **while** not converged
4. Solve $\mathbf{x}_k \in \mathbb{C}^n$ from $(s_kE - A)\mathbf{x}_k = E\mathbf{x}_{k-1}/\|\mathbf{x}_{k-1}\|_2$
5. Solve $\mathbf{y}_k \in \mathbb{C}^n$ from $(s_kE - A)^*\mathbf{y}_k = E^*\mathbf{y}_{k-1}/\|\mathbf{y}_{k-1}\|_2$
6. Compute the new pole estimate

$$s_{k+1} = \rho(\mathbf{x}_k, \mathbf{y}_k) = \frac{\mathbf{y}_k^* A \mathbf{x}_k}{\mathbf{y}_k^* E \mathbf{x}_k}$$

7. The pole $\lambda = s_{k+1}$ has converged if

$$\|A\mathbf{x}_k - s_{k+1}E\mathbf{x}_k\|_2 < \epsilon$$

8. Set $k = k + 1$
9. **end while**

4. Local convergence analysis. The generalized two-sided Rayleigh quotient (Definition 2.2) has some well-known basic properties (see [18, 20]):

- Homogeneity: $\rho(\alpha\mathbf{x}, \beta\mathbf{y}, \gamma A, \delta E) = (\gamma/\delta)\rho(\mathbf{x}, \mathbf{y}, A, E)$ for $\alpha, \beta, \gamma, \delta \neq 0$.
- Translation invariance: $\rho(\mathbf{x}, \mathbf{y}, A - \alpha E, E) = \rho(\mathbf{x}, \mathbf{y}, A, E) - \alpha$.
- Stationarity (all directional derivatives are zero): $\rho = \rho(\mathbf{x}, \mathbf{y}, A, E)$ is stationary if and only if $\mathbf{x}$ and $\mathbf{y}$ are right and left eigenvectors of (A, E), respectively, with eigenvalue ρ and $\mathbf{y}^*E\mathbf{x} \neq 0$.

4.1. Asymptotically quadratic rate of convergence. In [20, p. 689] it is proved that the asymptotic convergence rate of two-sided RQI is cubic for nondefective matrices. Along the same lines it can be shown that the asymptotic convergence rate of DPA is quadratic. For the eigenvalue, this also follows from the fact that DPA is an exact Newton method, but for the corresponding left and right eigenvectors the following lemma is needed, which gives a useful expression for $(\rho_{k+1} - \lambda)$ (using $s_{k+1} \equiv \rho_k \equiv \rho(\mathbf{x}_k, \mathbf{y}_k, A, E)$ from now on).

LEMMA 4.1. $\mathbf{v}$ $\mathbf{w}$ (A, E) λ $(A - \lambda E)\mathbf{v} = 0$ $\mathbf{w}^*(A - \lambda E) = 0$ $\mathbf{w}^*E\mathbf{v} = 1$ $\mathbf{w}^*\mathbf{b} \neq 0$ $\mathbf{c}^*\mathbf{x} \neq 0$ ρ_k (A, E) $\tau_k, \omega_k \in \mathbb{C}$ $\mathbf{x}_k$ $\mathbf{y}_k$

$$(\rho_kE - A)\mathbf{x}_k = \tau_k\mathbf{b} \qquad (\rho_kE - A)^*\mathbf{y}_k = \omega_k\mathbf{c} \tag{4.1}$$

$$\mathbf{x}_k = \mathbf{v} + \mathbf{d}_k \qquad \mathbf{y}_k = \mathbf{w} + \mathbf{e}_k, \tag{4.2}$$

$\mathbf{w}^*E\mathbf{d}_k = \mathbf{e}_k^*E\mathbf{v} = 0$ $\mathbf{u} \equiv (I - E\mathbf{v}\mathbf{w}^*)\frac{\mathbf{b}}{\mathbf{w}^*\mathbf{b}}$ $\mathbf{z} \equiv (I - E^*\mathbf{w}\mathbf{v}^*)\frac{\mathbf{c}}{\mathbf{v}^*\mathbf{c}}$

$$\mathbf{u} = (\rho_k - \lambda)^{-1}(\rho_kE - A)\mathbf{d}_k \perp \mathbf{w} \qquad \mathbf{z} = (\rho_k - \lambda)^{-*}(\rho_kE - A)^*\mathbf{e}_k \perp \mathbf{v},$$

[illegible] $\rho_{k+1} = \mathbf{y}_k^* A\mathbf{x}_k/(\mathbf{y}_k^* E\mathbf{x}_k)$ [illegible]

$$\rho_{k+1} - \lambda = (\rho_k - \lambda)\mu_k, \quad \text{[illegible]} \quad \mu_k = \frac{\mathbf{e}_k^* E\mathbf{d}_k - \mathbf{e}_k^*\mathbf{u}}{1 + \mathbf{e}_k^* E\mathbf{d}_k}. \tag{4.3}$$

[illegible] $\mathbf{u}$ [illegible] $\mathbf{z}$ [illegible]. Substitution of (4.2) into (4.1) and multiplication from the left by $\mathbf{w}^*$ and $\mathbf{v}^*$, respectively, give

$$\tau_k = \frac{\rho_k - \lambda}{\mathbf{w}^*\mathbf{b}} \quad \text{and} \quad \omega_k = \frac{(\rho_k - \lambda)^*}{\mathbf{v}^*\mathbf{c}}.$$

It follows that

$$(\rho_k E - A)\mathbf{d}_k = (\rho_k - \lambda)(I - E\mathbf{v}\mathbf{w}^*)\frac{\mathbf{b}}{\mathbf{w}^*\mathbf{b}} \equiv (\rho_k - \lambda)\mathbf{u} \perp \mathbf{w}$$

and

$$(\rho_k E - A)^*\mathbf{e}_k = (\rho_k - \lambda)^*(I - E^*\mathbf{w}\mathbf{v}^*)\frac{\mathbf{c}}{\mathbf{v}^*\mathbf{c}} \equiv (\rho_k - \lambda)^*\mathbf{z} \perp \mathbf{v},$$

where $\mathbf{u}$ and $\mathbf{z}$ are independent of the iteration. With $\rho_{k+1} = \mathbf{y}_k^* A\mathbf{x}_k/(\mathbf{y}_k^* E\mathbf{x}_k)$, it follows that

$$\rho_{k+1} - \lambda = \frac{\mathbf{y}_k^*(A - \lambda E)\mathbf{x}_k}{\mathbf{y}_k^* E\mathbf{x}_k} = \frac{\mathbf{e}_k^*(A - \lambda E)\mathbf{d}_k}{1 + \mathbf{e}_k^* E\mathbf{d}_k}.$$

Note that $\mathbf{e}_k^*(A-\lambda E)\mathbf{d}_k = \mathbf{e}_k^*(A-\rho_k E)\mathbf{d}_k + (\rho_k - \lambda)\mathbf{e}_k^* E\mathbf{d}_k = (\rho_k - \lambda)(\mathbf{e}_k^* E\mathbf{d}_k - \mathbf{e}_k^*\mathbf{u})$, which shows (4.3). □

This lemma will be used in the proof of the following theorem, which shows the asymptotically quadratic rate of convergence of DPA, and expression (4.3) in particular will be used to derive the local convergence neighborhoods of DPA and RQI in section 4.2.

THEOREM 4.2. [illegible] $\mathbf{v}$ [illegible] $\mathbf{w}$ [illegible] (A, E) [illegible] λ [illegible] $(A-\lambda E)\mathbf{v} = 0$ [illegible] $\mathbf{w}^*(A-\lambda E) = 0$ [illegible] $\mathbf{w}^* E\mathbf{v} = 1$ [illegible] $\lim_{k\to\infty} \mathbf{x}_k = \mathbf{v}$ [illegible] $\lim_{k\to\infty} \mathbf{y}_k = \mathbf{w}$ [illegible] $s_{k+1} = \rho_k = \rho(\mathbf{x}_k, \mathbf{y}_k)$ [illegible] λ [illegible]

[illegible]. The proof is an adaptation of the proofs in [20, p. 689] and [13, p. 150]. The main difference here is that for DPA the right-hand sides of the linear systems are kept fixed during the iterations. Let the iterates $\mathbf{x}_k$ and $\mathbf{y}_k$ (see Lemma 4.1) be of the form

$$\mathbf{x}_k = \mathbf{v} + \mathbf{d}_k \quad \text{and} \quad \mathbf{y}_k = \mathbf{w} + \mathbf{e}_k,$$

where $\mathbf{w}^* E\mathbf{d}_k = \mathbf{e}_k^* E\mathbf{v} = 0$ and $\mathbf{w}^* E\mathbf{v} = 1$. Put $\mathbf{d}_k = (\rho_k - \lambda)\tilde{\mathbf{d}}_k$ with $(\rho_k E - A)\tilde{\mathbf{d}}_k = \mathbf{u}$, and $\mathbf{e}_k = (\rho_k - \lambda)^*\tilde{\mathbf{e}}_k$ with $(\rho_k E - A)^*\tilde{\mathbf{e}}_k = \mathbf{z}$. Let $\mathcal{V}$ and $\mathcal{W}$ be the associated right and left eigenspaces for λ. Then there are two orthogonal decompositions of $\mathbb{C}^n$:

$$\mathbb{C}^n = \mathcal{V} \oplus (E^*\mathcal{W})^\perp = (E\mathcal{V})^\perp \oplus \mathcal{W},$$

and it can be shown that for all $z \in \mathbb{C}$, one has $(zE - A) : (E^*\mathcal{W})^\perp \longrightarrow \mathcal{W}^\perp$ and $(zE - A)^* : (E\mathcal{V})^\perp \longrightarrow \mathcal{V}^\perp$. Since these mappings are onto for all z sufficiently close to λ, there is a neighborhood $\mathcal{N}$ of λ and a constant $m > 0$ such that

$$\|(zE - A)\mathbf{s}\| \geq m\|\mathbf{s}\| \quad \text{and} \quad \|(zE - A)^*\mathbf{t}\| \geq m\|\mathbf{t}\|$$

for all $z \in \mathcal{N}$, $\mathbf{s} \in (E^*\mathcal{W})^\perp$, and $\mathbf{t} \in (E\mathcal{V})^\perp$. It follows that if $\rho_k \to \lambda$, then for sufficiently large k

$$\|\mathbf{d}_k\| \le \frac{|\rho_k - \lambda|}{m}\|\mathbf{u}\|, \tag{4.4}$$

and similarly

$$\|\mathbf{e}_k\| \le \frac{|\rho_k - \lambda|}{m}\|\mathbf{z}\|, \tag{4.5}$$

and $\mathbf{d}_k$ and $\mathbf{e}_k$, and $\tilde{\mathbf{d}}_k$ and $\tilde{\mathbf{e}}_k$, are bounded. Hence, $\mathbf{x}_k \to \mathbf{v}$ and $\mathbf{y}_k \to \mathbf{w}$ if $\rho_k \to \lambda$. The converse follows from the continuity of the Rayleigh quotient if and only if $\mathbf{x}_k \to \mathbf{v}$ and $\mathbf{y}_k \to \mathbf{w}$.

To prove the asymptotically quadratic rate of convergence, first note that

$$\rho_{k+1} - \lambda = \rho(\mathbf{x}_k, \mathbf{y}_k) = (\rho_k - \lambda)^2 \frac{\tilde{\mathbf{e}}_k^*(A - \lambda E)\tilde{\mathbf{d}}_k}{1 + (\rho_k - \lambda)^2 \tilde{\mathbf{e}}_k^* E \tilde{\mathbf{d}}_k},$$

and hence

$$|\rho_{k+1} - \lambda| = (\rho_k - \lambda)^2 |\tilde{\mathbf{e}}_k^*(A - \lambda E)\tilde{\mathbf{d}}_k| + O((\rho_k - \lambda)^4), \tag{4.6}$$

which proves that ρ_k converges quadratically toward λ.

The boundedness of $\tilde{\mathbf{d}}_k$ and $\tilde{\mathbf{e}}_k$ implies that, for ρ_k sufficiently close to λ,

$$|\rho_k - \lambda|^2 |\tilde{\mathbf{e}}_k^* E \mathbf{d}_k| \le \frac{1}{2}.$$

This leads to an estimate of $|\rho_{k+1} - \lambda|$ by $2|\rho_k - \lambda|^2 |\tilde{\mathbf{e}}_k^*(A - \lambda E)\tilde{\mathbf{d}}_k| = |\mathbf{e}_k^*(A - \lambda E)\mathbf{d}_k|$. A combination with (4.4) leads to

$$\begin{aligned} \|\mathbf{v} - \mathbf{x}_{k+1}\| &= \|\mathbf{d}_{k+1}\| \\ &\le \frac{\|A - \lambda E\|\ \|\mathbf{u}\|}{2m}\|\mathbf{v} - \mathbf{x}_k\|\|\mathbf{w} - \mathbf{y}_k\| \end{aligned}$$

for k sufficiently large, and similarly, a combination with (4.5) leads to

$$\begin{aligned} \|\mathbf{w} - \mathbf{y}_{k+1}\| &= \|\mathbf{e}_{k+1}\| \\ &\le \frac{\|A - \lambda E\|\ \|\mathbf{z}\|}{2m}\|\mathbf{v} - \mathbf{x}_k\|\|\mathbf{w} - \mathbf{y}_k\|, \end{aligned}$$

which proves the asymptotically quadratic convergence. □

4.2. Convergence neighborhood. In this section it will be assumed that A is a symmetric matrix and that $E = I$. In [17] Ostrowski characterizes the convergence neighborhood of the iteration

$$(A - \rho_k I)\mathbf{x}_k = \tau_k \mathbf{b}, \quad k = 0, 1, \ldots, \tag{4.7}$$

for symmetric matrices A, where ρ_0 is arbitrary, $\rho_{k+1} = \rho(\mathbf{x}_k, A) \equiv \rho(\mathbf{x}_k, \mathbf{x}_k, A, E)$ $(k > 0)$, and τ_k is a scalar so that $\|\mathbf{x}_k\|_2 = 1$. It can be seen that DPA for symmetric matrices (with $E = I$, $\mathbf{b} = \mathbf{c}$),

$$(\rho_k I - A)\mathbf{x}_k = \tau_k \mathbf{b}, \quad k = 0, 1, \ldots, \tag{4.8}$$

is similar, and hence Ostrowski's approach can be used to characterize the local convergence neighborhood of DPA for symmetric matrices A with $\mathbf{c} = \mathbf{b} = (b_1, \ldots, b_n)^T$. In fact, a larger convergence neighborhood of DPA will be derived here. This result gives insight into the typical convergence behavior of DPA.

Since the two-sided Rayleigh quotient and (4.7), (4.8) are invariant under unitary similarity transforms, without loss of generality A will be a diagonal matrix $\mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ with $\lambda_1 < \cdots < \lambda_n$. Note that $R_j = \|\mathbf{b}\|_2 \cos^2 \angle(\mathbf{v}_j, \mathbf{b})$ and that λ_J with $J = \mathrm{argmax}_j(\cos(\angle(\mathbf{v}_j, \mathbf{b})))$ is the dominant pole. The main results of this paper, sharp bounds for the convergence neighborhoods of DPA and RQI, respectively, are stated in Theorems 4.3 and 4.4, respectively. The proofs are given in section 4.2.1.

THEOREM 4.3. [illegible] $(\lambda, \mathbf{v})$ [illegible] A [illegible] A [illegible] $\mathbf{b}$ [illegible] ρ_0 [illegible] $\mathbf{x}_k$ [illegible] τ_k [illegible]

$$\|\mathbf{x}_k\| = 1, \quad (\rho_k I - A)\mathbf{x}_k = \tau_k \mathbf{b} \quad \text{[illegible]} \quad \rho_{k+1} \equiv \mathbf{x}_k^* A \mathbf{x}_k \quad (k \geq 0),$$

[illegible] $\gamma = \min_{\lambda_i \neq \lambda} |\lambda_i - \lambda|$ [illegible] $\alpha_k \equiv \frac{|\rho_k - \lambda|}{\gamma}$ [illegible] $\tilde{\alpha}_k \equiv \frac{\alpha_k}{1-\alpha_k}$ [illegible]

$$\alpha_0 < \alpha_{dpa} \equiv \frac{1}{1+\zeta^2} \quad \text{[illegible]} \quad \zeta \equiv \tan \angle(\mathbf{v}, \mathbf{b}), \tag{4.9}$$

[illegible] $\rho_k \to \lambda$ $(k \to \infty)$ [illegible] $k \geq 0$ [illegible]

$$\tilde{\alpha}_{k+1}\zeta^2 \leq \left(\tilde{\alpha}_k \zeta^2\right)^2 < 1.$$

With $c \equiv \cos \angle(\mathbf{v}, \mathbf{b})$ one has $c^2 = \frac{1}{1+\zeta^2}$, and hence condition (4.9) is equivalent to $\frac{|\rho_0 - \lambda|}{\gamma c^2} < 1$. In the setting of this paper, Ostrowski's convergence condition [17, equation (19), p. 235] is given by

$$\frac{|\rho_0 - \lambda|}{\gamma c^2} \leq \frac{1}{2} \min\left(\frac{1}{2(1-c^2)}, \frac{1}{c^2}\right).$$

Because $\frac{1}{2}\min(\frac{1}{2(1-c^2)}, \frac{1}{c^2}) \leq \frac{3}{4} < 1$, it is clear that the convergence neighborhood $\{\rho_0 \mid \text{condition (4.9) holds}\}$ that follows from Theorem 4.3 is larger than that from Ostrowski's condition.

In [17, p. 239], the convergence neighborhood of standard RQI,

$$(A - \rho_k I)\mathbf{x}_k = \tau_k \mathbf{x}_{k-1}, \quad k = 0, 1, \ldots, \tag{4.10}$$

where $\mathbf{x}_{-1}$ arbitrary, $\rho_{k+1} = \rho(\mathbf{x}_k, A)$ $(k > 0)$, and τ_k is a scalar so that $\|\mathbf{x}_k\|_2 = 1$, is derived. Here a sharper bound is derived.

THEOREM 4.4. [illegible] $(\lambda, \mathbf{v})$ [illegible] A [illegible] A [illegible] $\mathbf{b}$ [illegible] ρ_0 [illegible] $\mathbf{x}_{-1} = \mathbf{b}$ [illegible] $\mathbf{x}_k$ [illegible] τ_k [illegible]

$$\|\mathbf{x}_k\| = 1, \quad (\rho_k I - A)\mathbf{x}_k = \tau_k \mathbf{x}_{k-1} \quad \text{[illegible]} \quad \rho_{k+1} \equiv \mathbf{x}_k^* A \mathbf{x}_k \quad (k \geq 0),$$

[illegible] $\gamma = \min_{\lambda_i \neq \lambda} |\lambda_i - \lambda|$ [illegible] $\alpha_k \equiv \frac{|\rho_k - \lambda|}{\gamma}$, $\tilde{\alpha}_k \equiv \frac{\alpha_k}{1-\alpha_k}$, [illegible] $\zeta_{k+1} \equiv \tan \angle(\mathbf{v}, \mathbf{x}_k)$ [illegible]

$$\alpha_0 < \alpha_{rqi} \equiv \frac{1}{1+\zeta} \quad \zeta \equiv \zeta_0 \equiv \tan\angle(\mathbf{v}, \mathbf{b}), \tag{4.11}$$

$\zeta_1 < 1$, $|\rho_1 - \lambda| < \frac{1}{2}\gamma$, $\rho_k \to \lambda$ $(k \to \infty)$, $k \geq 0$,

$$\tilde{\alpha}_{k+1} \leq (\tilde{\alpha}_k \zeta_k)^2, \quad \zeta_{k+1} \leq \tilde{\alpha}_k \zeta_k, \quad \tilde{\alpha}_{k+1}\zeta_{k+1} \leq (\tilde{\alpha}_k \zeta_k)^3 \leq 1.$$

Note that the last inequality in the theorem also shows the cubic rate of convergence of RQI.

In [5, Thm. 1] it is shown that, with $\gamma_b = \beta - \alpha$ a known gap in the spectrum of A (for instance, $\gamma = \min_{\lambda_i \neq \lambda} |\lambda - \lambda_i|$), if $\rho_1 < (\alpha+\beta)/2$ $\|\mathbf{r}_1\| = \|A\mathbf{x}_0 - \rho_1 \mathbf{x}_0\| \leq \gamma_b$, then $\rho_k < (\alpha+\beta)/2$ for $k \geq 1$, and similarly for the case $\rho_1 > (\alpha+\beta)/2$. The first condition of this theorem implies that $|\rho_1 - \lambda| < \gamma/2$, while $\|\mathbf{r}_1\| \leq \gamma$ is possible only if $\angle(\mathbf{v}, \mathbf{x}_0) < 45°$. In other words, $\zeta < 1$ and $|\rho_1 - \lambda|/\gamma < 1/2 < 1/(1+\zeta)$. As can be learned from the proof of Theorem 4.4 in section 4.2.1, this is the situation after one iteration step. Theorem 4.4 seems to allow a weaker start. To see this, consider the two-dimensional example $A = \mathrm{diag}(-1, 1)$. With $\rho_0 = 0.01$, $\mathbf{x}_{-1} = \mathbf{b} = [\sqrt{2}/2, \sqrt{2}/2]$, and $\mathbf{x}_0 = (A - \rho_0 I)^{-1}\mathbf{x}_{-1}$, it follows that $|\lambda - \rho_0| < 1$ and condition (4.11) is satisfied, while $\|\mathbf{r}_1\| = \|A\mathbf{x}_0 - \rho_1\mathbf{x}_0\| \approx 1.03 > 1$. Hence, the result in Theorem 4.4 is sharper.

The results of Theorems 4.3 and 4.4 are sharp in the following sense.

THEOREM 4.5. 4.3 4.4 $\mathbf{b}$ ρ_0 λ λ_{i_0} ρ_k 4.3 (4.4) λ $|\rho_0 - \lambda| < \alpha_{dpa}$ $(< \alpha_{rqi}$) i_0 $\gamma = |\lambda_{i_0} - \lambda|$

Let $\mathbf{b} = c\mathbf{v} + c_j \mathbf{v}_{i_0}$. For ease of notation, write $(\lambda_1, \mathbf{v}_1) = (\lambda, \mathbf{v})$ and $(\lambda_2, \mathbf{v}_2) = (\lambda_{i_0}, \mathbf{v}_{i_0})$.

With $\alpha_0^i \equiv |\rho_0 - \lambda_i|/\gamma$ and $\zeta_0^{(i)} \equiv \tan\angle(\mathbf{v}_i, \mathbf{b})$ $(i = 1, 2)$, one has that $\alpha_0^{(2)} = 1 - \alpha_0^{(1)}$ and $\zeta_0^{(1)} = 1/\zeta_0^{(2)}$. Therefore, if $\alpha_0^{(1)} > \gamma/(1 + (\zeta_0^{(1)})^p)$, then $\alpha_0^{(2)} < \gamma/(1 + (\zeta_0^{(2)})^p)$ $(p = 1, 2)$, and Theorem 4.3 (take $p = 2$) and Theorem 4.4 (take $p = 1$) guarantee convergence toward λ_2.

If $\alpha_0^{(1)} = \gamma/(1 + (\zeta_0^{(1)})^p)$, then $\alpha_0^{(2)} = \gamma/(1 + (\zeta_0^{(2)})^p)$, and, as can be seen in the proof of the corresponding theorem, the contraction statement in the theorem holds for both $\lambda = \lambda_1$ and $\lambda = \lambda_2$. This implies stagnation of the sequence of $|\rho_k - \lambda|$. ∎

Note that it is actually proved that Theorem 4.5 is correct for any nontrivial $\mathbf{b}$ in the two-dimensional subspace spanned by $\mathbf{v}$ and $\mathbf{v}_j$ with j such that $\gamma = |\lambda_j - \lambda|$.

In Figure 4.1, α_{dpa} and α_{rqi} (see equations (4.9) and (4.11), respectively) are plotted for c_j^2, where $c_j = \cos\angle(\mathbf{v}, \mathbf{b})$. As c_j^2 increases, i.e., as mode j becomes more dominant, both local convergence neighborhoods increase and $\alpha \to 1$, while the bound for the DPA neighborhood is larger for $c_j^2 > 1/2$, or $\angle(\mathbf{v}, \mathbf{b}) < 45°$.

The price one has to pay for the cubic convergence is the smaller local convergence neighborhood of the dominant pole, as it becomes more dominant, for RQI. While DPA emphasizes the dominant mode at every iteration by keeping the right-hand side fixed, RQI takes advantage of this only in the first iteration, and for initial shifts too far from the dominant pole, the dominant mode may be damped out from the iterates $\mathbf{x}_k$. In that sense, RQI is closer to the inverse power method or inverse iteration, which converges to the eigenvalue closest to the shift, while DPA takes advantage of the information in the right-hand side $\mathbf{b}$.

Because the results are in fact lower bounds for the local convergence neighborhood, theoretically speaking, no conclusions can be drawn about the global basins of attraction. But the results strengthen the intuition that for DPA the basin of attraction of the dominant pole is larger than that for RQI.

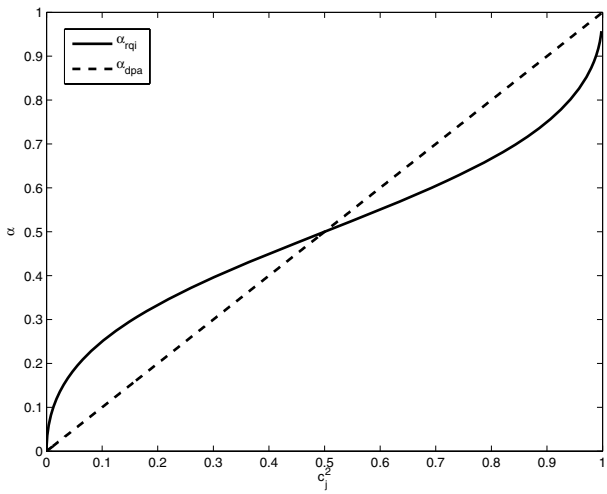


FIG. 4.1. *Bounds of the local convergence neighborhood for DPA (dashed) and best-case RQI (solid). If, with* $\gamma = \min_{i \neq j} |\lambda_i - \lambda_j|$, *one has* $|\lambda_j - \rho_\ell| \leq \alpha\gamma$ *for some* ℓ, *then the sequence of* ρ_k *converges to* λ_j. *Here,* $\alpha = \alpha_{dpa}$ *or* $\alpha = \alpha_{rqi}$. *Along the horizontal axis* c_j^2 *is varied between* 0 *(***b** *is orthogonal to* $\mathbf{v}_j$*) and* 1 *(***b** *is a multiple of* $\mathbf{v}_j$*):* $c_j \equiv \cos\angle(\mathbf{v}_j, \mathbf{b})$.

4.2.1. Proofs of Theorems 4.3 and 4.4. The following two lemmas provide expressions and bounds that are needed for the proofs of Theorems 4.3 and 4.4.

LEMMA 4.6. [illegible] $\mathbf{v}$ [illegible] $A = A^T$ [illegible] λ [illegible] $\|\mathbf{v}\| = 1$, [illegible] $\tau_k \in \mathbb{R}$ [illegible] $\mathbf{x}_k$ [illegible]

$$(\rho_k I - A)\mathbf{x}_k = \tau_k \mathbf{b}$$

[illegible]

$$\mathbf{x}_k = \mathbf{v} + \mathbf{d}_k, \tag{4.12}$$

[illegible] $\mathbf{v}^*\mathbf{d}_k = 0$ [illegible] $\mathbf{z} = (\rho_k - \lambda)^{-1}(\rho_k I - A)\mathbf{d}_k$ [illegible] $\rho_{k+1} = \mathbf{x}_k^* A\mathbf{x}_k / (\mathbf{x}_k^*\mathbf{x}_k)$ [illegible]

$$\rho_{k+1} - \lambda = (\rho_k - \lambda)\mu_k,$$

[illegible]

$$\mu_k = \frac{\mathbf{d}_k^*\mathbf{d}_k - \mathbf{d}_k^*\mathbf{z}}{1 + \mathbf{d}_k^*\mathbf{d}_k}. \tag{4.13}$$

[illegible]. The result follows from Lemma 4.1, by noting that $A = A^T$ and $E = I$. □

LEMMA 4.7. [illegible] 4.6 [illegible] $\gamma = \min_{\lambda_i \neq \lambda} |\lambda_i - \lambda|$, $c = \cos\angle(\mathbf{v}, \mathbf{b})$, $\zeta = \|\mathbf{z}\|$, $\alpha_k = \frac{|\rho_k - \lambda|}{\gamma}$ [illegible] $\tilde{\alpha}_k = \alpha_k/(1 - \alpha_k)$ [illegible] [illegible]

$$\alpha_k < 1 \quad \Rightarrow \quad \|\mathbf{d}_k\| \leq \tilde{\alpha}_k \zeta, \tag{4.14}$$

$\alpha_k \le c = 1/\sqrt{1+\zeta^2}$

$$\alpha_k < 1 \quad \Rightarrow \quad \tilde{\alpha}_{k+1} \equiv \frac{\alpha_{k+1}}{1-\alpha_{k+1}} \le (\tilde{\alpha}_k \zeta)^2. \tag{4.15}$$

Put $\zeta_k = \|\mathbf{d}_k\|$. Then by (4.13)

$$|\mu_k| \le \phi(\zeta_k), \text{ where } \phi(\tau) \equiv \frac{\zeta\tau + \tau^2}{1+\tau^2} \quad (\tau \in \mathbb{R}).$$

The function ϕ is increasing on $[0, \tau_{\max}]$, where $\tau_{\max} = (1+\sqrt{1+\zeta^2})/\zeta$, or, using $c \equiv \cos\angle(\mathbf{v}, \mathbf{b}) = 1/\sqrt{1+\zeta^2}$, $\tau_{\max} = \sqrt{(1+c)/(1-c)}$, and $0 \le \phi \le \frac{1+c}{2c}$ on $(0, \infty)$.

Since $\|(A-\rho_k)^{-1}|_{\mathbf{v}^\perp}\| \le |1/(\gamma - |\lambda - \rho_k|)|$, it follows that

$$\zeta_k \equiv \|\mathbf{d}_k\| \le |\rho_k - \lambda| \|(A-\rho_k)^{-1}|_{\mathbf{v}^\perp}\| \|\mathbf{z}\| \le \frac{|\rho_k - \lambda|}{|\gamma - |\rho_k - \lambda||} = \frac{\alpha_k}{1-\alpha_k}\zeta,$$

which proves (4.14). The statement that if $\alpha_k \le c = 1/\sqrt{1+\zeta^2}$, then $\zeta_k \le \tilde{\alpha}_k\zeta \le \tau_{\max}$ and

$$|\mu_k| \le \phi(\zeta_k) \le \phi(\tilde{\alpha}_k\zeta) \le \frac{\tilde{\alpha}_k + \tilde{\alpha}_k^2}{1+\tilde{\alpha}_k^2\zeta^2}\zeta^2 = \frac{\alpha_k\zeta^2}{(1-\alpha_k)^2 + \alpha_k^2\zeta^2} \tag{4.16}$$

now follows from the observation that $\alpha_k\zeta/(1-\alpha_k) \le \tau_{\max}$ if and only if $\alpha_k \le 1/\sqrt{1+\zeta^2} = c$. Furthermore, if $\alpha_k \le c = 1/\sqrt{1+\zeta^2}$, then

$$|\mu_k| \le 1 \quad \text{if} \quad \tilde{\alpha}_k\zeta^2 \le 1 \quad \left(\Leftrightarrow \quad \alpha_k \le \frac{1}{1+\zeta^2} = c^2\right). \tag{4.17}$$

This follows readily from $\phi(\tilde{\alpha}_k\zeta) \le 1$, statement (4.14), and the definition $\tilde{\alpha}_k = \alpha_k/(1-\alpha_k)$.

Finally, statement (4.15) follows from the fact that (4.16) and (4.15) imply $\alpha_{k+1} \le (\tilde{\alpha}_k\zeta)^2/(1+(\tilde{\alpha}_k\zeta)^2)$. □

Note that it is essential that the function ϕ is increasing, since this allows us to use upper bound (4.14) to also handle the denominator in (4.13), leading to (4.16).

In the two-dimensional case, the estimate in (4.16) is sharp (equality), since both $\mathbf{z}$ and $\mathbf{d}_k$ are in the same direction (orthogonal to $\mathbf{v}$). Furthermore, in (4.17), $|\mu| \le 1$ if and only if $\tilde{\alpha}_k\zeta^2 \le 1$.

4.3. Note that $\zeta \equiv \zeta_0$ is the same in all iterations, and recall that $\alpha_k \equiv |\rho_k - \lambda|/\gamma$. Since $c^2 = 1/(1+\zeta^2)$, condition (4.9) implies $\alpha_0(1+\zeta^2) < 1$, and by induction and (4.15) of Lemma 4.7, it follows that

$$\tilde{\alpha}_{k+1}\zeta^2 \le (\tilde{\alpha}_k\zeta^2)^2 \quad \text{if} \quad \tilde{\alpha}_0\zeta^2 \le 1 \qquad (k \ge 0), \tag{4.18}$$

which implies convergence if $\tilde{\alpha}_0\zeta_0^2 < 1$. □

4.4. Note that $\zeta_{k+1} = \tan\angle(\mathbf{v}, \mathbf{x}_k)$ changes every iteration, and recall that $\alpha_k \equiv |\rho_k - \lambda|/\gamma$ and $\tilde{\alpha}_k = \alpha_k/(1-\alpha_k)$. Condition (4.11) implies $\alpha_0 < 1/(1+\zeta)$ or, equivalently, $\tilde{\alpha}_0\zeta_0 < 1$. By (4.15) it follows that $\zeta_1 < 1$ and $\alpha_1 < 1/2$, or, equivalently, $|\rho_1 - \lambda| < \gamma/2$, as announced in the discussion following Theorem 4.4. Since $\tilde{\alpha}_k\zeta_k < 1$ implies that $\alpha_k < 1/\sqrt{1+\zeta_k^2}$, results (4.14) and (4.15) of Lemma 4.7 can be applied to obtain

$$\tilde{\alpha}_{k+1} \le (\tilde{\alpha}_k\zeta_k)^2 \quad \text{and} \quad \tilde{\alpha}_{k+1}\zeta_{k+1} \le (\tilde{\alpha}_k\zeta_k)^3$$

if $\tilde{\alpha}_k\zeta_k < 1$. Here it is used that (4.14) reads in this context as $\zeta_{k+1} \le \tilde{\alpha}_k\zeta_k$. Since $\tilde{\alpha}_0\zeta_0 < 1$, an induction argument shows the cubic rate of convergence. □

4.3. General systems. Theorems 4.3 and 4.4 can readily be generalized for normal matrices, but it is difficult to obtain such bounds for general matrices without making specific assumptions. To see this, note that it is difficult to give sharp bounds for (4.3) in Lemma 4.1. However, the following theorem states that DPA is invariant under certain transformations and helps in getting more insight into DPA for general, nondefective systems $(A, E, \mathbf{b}, \mathbf{c})$.

THEOREM 4.8. [illegible] (A, E) [illegible] $X, Y \in \mathbb{C}^{n\times n}$ [illegible] $(A, E, \mathbf{b}, \mathbf{c}, s_0)$ [illegible] $(\mathbf{x}_k, \mathbf{y}_k, s_{k+1})$ [illegible] $(Y^*AX, Y^*EX, Y^*\mathbf{b}, X^*\mathbf{c}, s_0)$ [illegible] $(X^{-1}\mathbf{x}_k, Y^{-1}\mathbf{y}_k, s_{k+1})$ [illegible]. If $\mathbf{x} = \mathbf{x}_k$ is the solution of

$$(sE - A)\mathbf{x} = \mathbf{b},$$

then $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}_k = X^{-1}\mathbf{x}$ is the solution of

$$(sY^*EX - Y^*AX)\tilde{\mathbf{x}} = Y^*\mathbf{b},$$

and vice versa. Similar relations hold for $\mathbf{y} = \mathbf{y}_k$ and $\tilde{\mathbf{y}} = \tilde{\mathbf{y}}_k = Y^{-1}\mathbf{y}$. Noting that

$$s_{k+1} = \frac{\tilde{\mathbf{y}}^*Y^*AX\tilde{\mathbf{x}}}{\tilde{\mathbf{y}}^*Y^*EX\tilde{\mathbf{x}}} = \frac{\mathbf{y}^*A\mathbf{x}}{\mathbf{y}^*E\mathbf{x}} = s_{k+1}$$

completes the proof. □

Let W and V have as their columns the left and right eigenvectors of (A, E), respectively, i.e., $AV = EV\Lambda$ and $W^*A = \Lambda W^*E$, with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. Furthermore, let W and V be scaled so that $W^*EV = \Delta$, where Δ is a diagonal matrix with $\delta_{ii} = 1$ for finite λ_i and $\delta_{ii} = 0$ for $|\lambda_i| = \infty$. According to Theorem 4.8, DPA$(A, E, \mathbf{b}, \mathbf{c})$ and DPA$(\Lambda, \Delta, W^*\mathbf{b}, V^*\mathbf{c})$ produce the same pole estimates s_k. In $\tilde{\mathbf{b}} = W^*\mathbf{b}$ and $\tilde{\mathbf{c}} = V^*\mathbf{c}$, the new right-hand sides, one recognizes the contributions to the residues $R_i = \tilde{\mathbf{c}}_i\tilde{\mathbf{b}}_i = (\mathbf{c}^*\mathbf{v}_i)(\mathbf{w}_i^*\mathbf{b})$. The more dominant pole λ_i is, the larger the corresponding coefficients $\tilde{\mathbf{b}}_i$ and $\tilde{\mathbf{c}}_i$ are, and, since (Λ, Δ) is a diagonal pencil, the larger the chance that DPA converges to the unit vectors $\tilde{\mathbf{v}} = \mathbf{e}_i$ and $\tilde{\mathbf{w}} = \mathbf{e}_i$, which correspond to the right and left eigenvectors $\mathbf{v}_i = V\mathbf{e}_i$ and $\mathbf{w}_i = W\mathbf{e}_i$, respectively.

As observed earlier, DPA emphasizes the dominant mode every iteration by keeping the right-hand sides fixed and thereby can be expected to enlarge the convergence neighborhood also for general systems, compared to two-sided RQI. In practice, the quadratic instead of cubic rate of local convergence costs at most 2 or 3 iterations. Numerical experiments confirm that the basins of attraction of the dominant eigenvalues are larger for DPA, as will be discussed in the following section.

5. Basins of attraction and typical convergence behavior. It is not straightforward to characterize the global convergence of DPA, not even for symmetric matrices (see [17, pp. 236–237]). Basins of attraction of RQI in the three-dimensional case are studied in [1, 4, 19], while in [5, 27] local convergence neighborhoods are described. Because the DPA residuals $\mathbf{r}_k = (A - \rho_k I)\mathbf{b}$ are not monotonically decreasing (in contrast to the inverse iteration residuals $\mathbf{r}_k = (A - \sigma I)\mathbf{x}_k$ and the RQI residuals $\mathbf{r}_k = (A - \rho_k I)\mathbf{x}_k$; see [5, 19, 20]), it is not likely that similar results can be obtained for DPA. Numerical experiments, however, may help us to get an idea of the typical convergence behavior of DPA and may show why DPA is to be preferred over two-sided RQI for the computation of dominant poles.

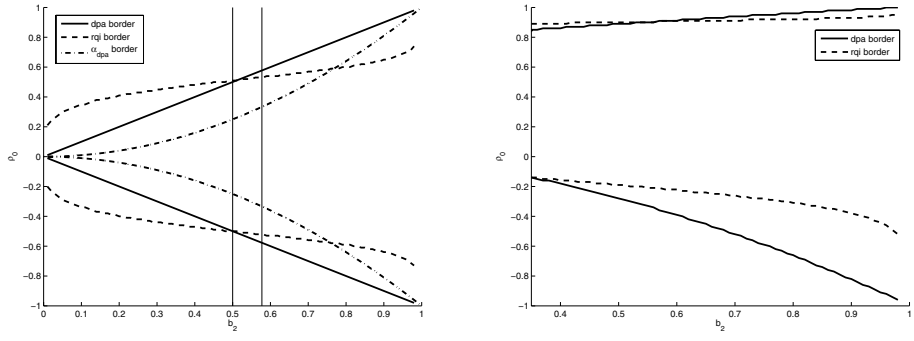


FIG. 5.1. *Convergence regions for DPA (solid borders) and RQI (dashed), and the theoretical DPA border (dash-dot; see Theorem 4.3) for the matrix* $A = diag(-1, s, 1)$ *for* $s = 0$ *(left) and* $s = 0.8$. *The regions of convergence to* $\lambda_2 = s$ *for DPA and RQI, respectively, are enclosed between the lower and upper borders of DPA and RQI, respectively. The regions of convergence to* $\lambda_1 = -1$ *(*$\lambda_3 = 1$*) are below (above) the lower (upper) border.*

An unanswered question is how to choose the initial shift of DPA. An obvious choice is the two-sided Rayleigh quotient $s_0 = (\mathbf{c}^* A\mathbf{b})/(\mathbf{c}^* E\mathbf{b})$. This choice will work in the symmetric case $A = A^*, E = I, \mathbf{c} = \mathbf{b}$. In the general nonsymmetric case this choice will not always be possible: the vectors $\mathbf{b}$ and $\mathbf{c}$ are often very sparse (only $O(1)$ nonzero entries), and, moreover, it may happen that $\mathbf{c}^* E\mathbf{b} = 0$. In that case the initial shift should be based on heuristics. For two-sided RQI, an obvious choice is to take as the initial vectors $\mathbf{x}_0 = \mathbf{b}$ and $\mathbf{y}_0 = \mathbf{c}$, but similarly, if $\mathbf{y}_0^* E\mathbf{x}_0 = 0$, this fails. Therefore, in the following experiments an initial shift s_0 will be chosen and the (normalized) initial vectors for two-sided RQI are $\mathbf{x}_0 = (A - s_0 E)^{-1}\mathbf{b}$ and $\mathbf{y}_0 = (A - s_0 E)^{-1}\mathbf{c}$; see Algorithm 2.

All experiments were executed in MATLAB 7 [28]. The criterion for convergence was $\|A\mathbf{x}_k - s_{k+1} E\mathbf{x}_k\|_2 < 10^{-8}$.

5.1. Three-dimensional symmetric matrices. Because RQI and DPA are shift and scaling invariant, the region of all 3×3 symmetric matrices can be parametrized by $A = \mathrm{diag}(-1, s, 1)$, with $0 \le s < 1$ due to symmetry (see [19]). In order to compute the regions of convergence of RQI and DPA (as defined in (4.7), (4.8)), the algorithms are applied to A for initial shifts in the range $(-1, 1)\backslash\{s\}$, with $\mathbf{c} = \mathbf{b} = (b_1, b_2, b_3)^T$, where $0 < b_2 \le 1$ and $b_1 = b_3 = \sqrt{(1 - b_2^2)/2}$. In Figure 5.1 the results are shown for $s = 0$ and $s = 0.8$. The intersections $\rho = \rho_{\lambda_1}$ and $\rho = \rho_{\lambda_3}$ at $b_2 = b$ with the borders define the convergence regions: for $-1 \le \rho_0 < \rho_{\lambda_1}$ there is convergence to $\lambda_1 = -1$, for $\rho_{\lambda_1} \le \rho_0 < \rho_{\lambda_3}$ there is convergence to $\lambda_2 = s$, while for $\rho_{\lambda_3} \le \rho_0 \le 1$ there is convergence to $\lambda_3 = 1$.

For the case $s = 0$ it can be observed that (see vertical lines) for $0 \le b_2 \lesssim 0.5$, the convergence region to the dominant extremal eigenvalues is larger for DPA. For $0.5 \lesssim b_2 \le 1/\sqrt{3} \approx 0.577$, the point at which λ_2 becomes dominant, the convergence region of RQI is larger. However, for $b_2 \gtrsim 0.5$, the convergence region of λ_2 is clearly larger for DPA. Note also that the theoretical (lower bound $\alpha_{dpa}\gamma$ of the) local convergence neighborhood for DPA (Theorem 4.3) is even larger than the practical convergence neighborhood of two-sided RQI for $b_2 \gtrsim 0.8$.

A similar observation can be made for the case $s = 0.8$. There, due to the decentralized location of λ_2, the figure is not symmetric and the region of convergence

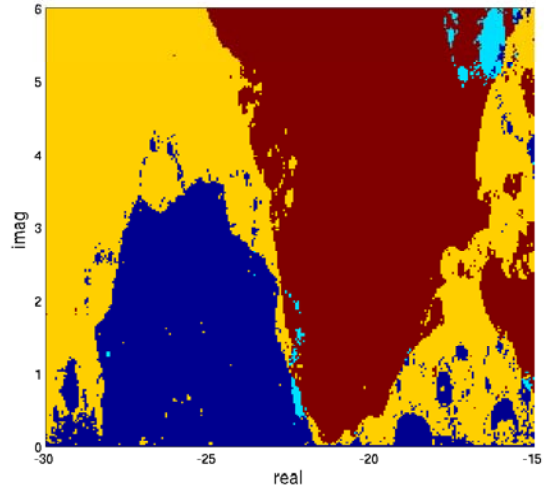


FIG. 5.2. *Convergence regions for DPA and two-sided RQI for the example of section* 5.2. *The center of the domain is the pole* $\lambda \approx -20.5 \pm 1.1i$, *with residue norm* $|R| \approx 6.2 \cdot 10^{-3}$. *Yellow and red points (light-blue and red points) mark initial shifts for which convergence to the target takes place for DPA (two-sided RQI); dark-blue points denote convergence to less dominant poles. Real (imaginary) parts of initial shifts are at the horizontal (vertical) axis. Horizontal and vertical strides are* 15/250 *and* 6/250, *respectively.*

of λ_2 is clearly larger for DPA. For $0 \leq b_2 \lesssim 0.35$, DPA and RQI appear to be very sensitive to the initial shift. While the convergence region for λ_1 was similar to the case $s = 0$, convergence for $-0.1 \lesssim \rho_0 \lesssim 0.8$ was irregular in the sense that for initial shifts in this interval both λ_2 and λ_3 could be computed; hence the regions are shown only for $b_2 \gtrsim 0.35$. Because the theoretical lower bounds are much smaller, since $d = \min_{i \neq j} |\lambda_i - \lambda_j| = 0.2$, and make the figure less clear, they are not shown (the theoretical DPA border still crosses the practical two-sided RQI border around $b_2 \approx 0.9$).

It is almost generic that, apart from a small interval of values of b_2, the area of convergence of the dominant eigenvalue is larger for DPA than for RQI. The following example discusses a large scale general system.

5.2. A large scale example. This example is a test model of the Brazilian Interconnect Power System (BIPS) [23, 22]. The sparse matrices A and E are of dimension $n = 13,251$ and E is singular. The input and output vectors $\mathbf{b}$ and $\mathbf{c}$ have only one nonzero entry, and, furthermore, $\mathbf{c}^T E \mathbf{b} = 0$; the choice $\mathbf{x}_0 = \mathbf{b}$ and $\mathbf{y}_0 = \mathbf{c}$ is not practical; see the beginning of this section. The pencil (A, E) is nonnormal and the most dominant poles appear in complex conjugated pairs. It is not feasible to determine the convergence regions for the entire complex plane, but the convergence behavior in the neighborhood of a dominant pole can be studied by comparing the found poles for a number of initial shifts in the neighborhood of the pole, for both DPA and two-sided RQI (Algorithms 1 and 2). The result is shown in Figure 5.2.

Initial shifts for which DPA and two-sided RQI converge to the target (the most dominant pole $\lambda \approx -20.5 \pm 1.1i$) or its complex conjugate are marked by yellow and red points, and light-blue and red points, respectively. Red grid points denote convergence to the most dominant pole for both DPA and two-sided RQI. Dark-blue grid points denote convergence to a less dominant pole.

In Figure 5.2 the target is the most dominant pole of the system. It can be clearly observed that for DPA the number of initial shifts that converge to the dominant pole (yellow and red points) is larger than for two-sided RQI (light-blue and red points). The basin of attraction of the dominant pole is larger for DPA: except for regions in the neighborhood of other relatively dominant poles (see, for instance, the poles in the interval $(-28, -24)$ on the real axis), there is convergence to the most dominant pole. For DPA typically the size of the basin of attraction increases with the relative dominance of the pole, while for two-sided RQI the effect is less strong; cf. Theorem 4.3, Theorem 4.4, and the discussion in section 4.2. The figure is symmetric with respect to the real axis: if for initial shift s_0, DPA (two-sided RQI) produces the sequence $(\mathbf{x}_k, \mathbf{y}_k, s_{k+1})$ converging to $(\mathbf{v}, \mathbf{w}, \lambda)$, then for $\bar{s}_0$ it produces the sequence $(\bar{\mathbf{x}}_k, \bar{\mathbf{y}}_k, \bar{s}_{k+1})$ converging to $(\bar{\mathbf{v}}, \bar{\mathbf{w}}, \bar{\lambda})$.

It can be seen that for many initial shifts (yellow points) DPA converges to the most dominant pole, but two-sided RQI does not. On the other hand, for a very small number of initial shifts (light-blue points), two-sided RQI converges to the most dominant pole while DPA does not. This is a counterexample for the obvious thought that if two-sided RQI converges to the dominant pole, then DPA also converges to it.

The average number of iterations needed by DPA to converge to the most dominant pole was 7.2, while two-sided RQI needed an average number of 6.0 iterations. The average numbers over the cases where both DPA and two-sided RQI converged to the most dominant pole were 6.1 and 5.9 iterations, respectively.

Similar behavior is observed for other systems and transfer functions. Although the theoretical and experimental results do not provide hard evidence in the sense that they prove that the basin of attraction of the dominant pole is larger for DPA than for two-sided RQI, they at least indicate an advantage of DPA over two-sided RQI.

5.3. PEEC example. The PEEC system [8] is a well-known benchmark system for model order reduction applications. One of the difficulties with this system of order $n = 480$ is that it has many equally dominant poles that lie close to each other in a relatively small part, $[-1, 0] \times [-10i, 10i]$, of the complex plane. This explains why in Figure 5.3 for only a relatively small part of the plane there is convergence (marked by yellow and red, and light-blue and red, for DPA and two-sided RQI, respectively) to the most dominant pole $\lambda \approx -0.14 \pm 5.4i$.

Although the difference is less pronounced than in the previous examples, DPA still converges to the most dominant pole in more cases than two-sided RQI, and the average residue norm of the found poles was also larger: $R_{avg}^{dpa} \approx 5.2 \cdot 10^{-3}$ versus $R_{avg}^{rqi} \approx 4.5 \cdot 10^{-3}$. Again a remarkable observation is that even for some initial shifts very close to another pole, DPA converges to the most dominant pole, while two-sided RQI converges to the nearest pole; e.g., for initial shift $s_0 = 5i$ DPA converges to the most dominant pole $\lambda \approx -0.143 + 5.38i$ with $|R| \approx 7.56 \cdot 10^{-3}$, while two-sided RQI converges to the less dominant pole $\lambda \approx -6.3 \cdot 10^{-3} + 4.99i$ with $|R| \approx 3.90 \cdot 10^{-5}$.

The average number of iterations needed by DPA to converge to the most dominant pole was 9.8, while two-sided RQI needed an average number of 7.9 iterations. The average numbers over the cases where both DPA and two-sided RQI converged to the most dominant pole were 9.4 and 7.7 iterations, respectively.

Another nice observation is the appearance of fractal boundaries, typical for Newton processes.

6. Conclusions. The theoretical and numerical results confirm the intuition, and justify the conclusion, that the dominant pole algorithm (DPA) has better global

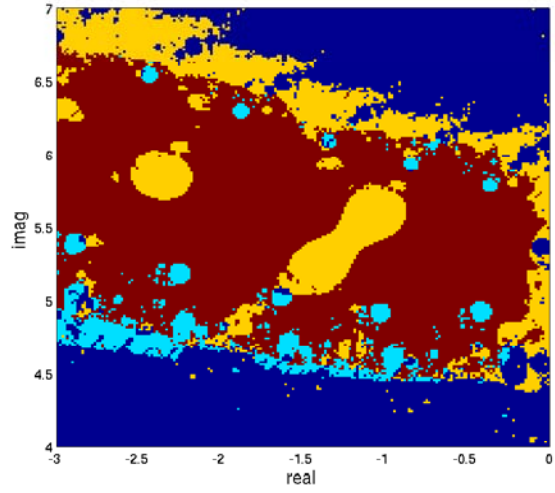


FIG. 5.3. *Convergence regions for DPA and two-sided RQI. The target is the dominant pole* $\lambda \approx -0.14 \pm 5.4i$*, with residue norm* $|R| \approx 7.6 \cdot 10^{-3}$*. Yellow and red points (light-blue and red points) mark initial shifts for which convergence to the target takes place for DPA (two-sided RQI); dark-blue points denote convergence to less dominant poles. Real (imaginary) parts of initial shifts are at the horizontal (vertical) axis. Horizontal and vertical strides are both* 3/200.

convergence than two-sided Rayleigh quotient iteration (RQI) to the dominant poles of a large scale dynamical system. The derived local convergence neighborhoods of dominant poles are larger for DPA, as the poles become more dominant, and numerical experiments indicate that the local basins of attraction of the dominant poles are larger for DPA than for two-sided RQI.

Both DPA and two-sided RQI need to solve two linear systems at every iteration. The difference between DPA and two-sided RQI is that DPA keeps the right-hand sides fixed to the input and output vectors of the system, while two-sided RQI updates the right-hand sides at every iteration. The more dominant a pole is, the bigger the difference in convergence behavior between DPA and two-sided RQI. The other way around, for considerably less dominant poles, the basins of attraction are much smaller for DPA than for two-sided RQI. This could be observed in cases where the initial shift was very close to a less dominant pole and DPA converged to a more dominant pole, while two-sided RQI converged to the nearest, less dominant pole.

The fact that DPA has an asymptotically quadratic rate of convergence, against a cubic rate for two-sided RQI, is of minor importance, since this has only a very local effect and hence leads to a small difference in the number of iterations (typically a difference of 1 or 2 iterations).

Acknowledgments. We are grateful to Nelson Martins for fruitful discussions about DPA [15] and its follow-ups SADPA [23] and SAMDP [22]. He also provided us with the New England and BIPS test systems. We thank Henk van der Vorst for useful comments on earlier versions of this paper and for suggestions that helped us to improve the presentation of the paper. Finally, we are much indebted to the anonymous referees, whose detailed comments and suggestions led to improved readability and presentation of the paper.

REFERENCES

[1] P.-A. Absil, R. Sepulchre, P. Van Dooren, and R. Mahony, *Cubically convergent iterations for invariant subspace computation*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 70–96.

[2] L. A. Aguirre, *Quantitative measure of modal dominance for continuous systems*, in Proceedings of the 32nd IEEE Conference on Decision and Control, IEEE Control Systems Society, Piscataway, NJ, 1993, pp. 2405–2410.

[3] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, Adv. Des. Control 6, SIAM, Philadelphia, 2005.

[4] S. Batterson and J. Smillie, *The dynamics of Rayleigh quotient iteration*, SIAM J. Numer. Anal., 26 (1989), pp. 624–636.

[5] C. Beattie and D. W. Fox, *Localization criteria and containment for Rayleigh quotient iteration*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 80–93.

[6] L. H. Bezerra, *Written discussion to* [15], IEEE Trans. Power Syst., 11 (1996), p. 168.

[7] L. H. Bezerra, *An eigenvalue method for calculating dominant poles of a transfer function*, Appl. Math. Lett., 21 (2008), pp. 244–247.

[8] Y. Chahlaoui and P. van Dooren, *A Collection of Benchmark Examples for Model Reduction of Linear Time Invariant Dynamical Systems*, SLICOT Working Note 2002-2, National Institute for Research and Development in Informatics, Bucharest, Romania, 2002.

[9] V. Frayssé and V. Toumazou, *A note on the normwise perturbation theory for the regular generalized eigenproblem* $Ax = \lambda Bx$, Numer. Linear Algebra Appl., 5 (1998), pp. 1–10.

[10] M. Green and D. J. N. Limebeer, *Linear Robust Control*, Prentice–Hall, Englewood Cliffs, NJ, 1995.

[11] A. M. A. Hamdan and A. H. Nayfeh, *Measures of modal controllability and observability for first- and second-order linear systems*, J. Guidance Control Dynam., 12 (1989), pp. 421–428.

[12] D. J. Higham and N. J. Higham, *Structured backward error and condition of generalized eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 493–512.

[13] M. E. Hochstenbach and G. L. G. Sleijpen, *Two-sided and alternating Jacobi-Davidson*, Linear Algebra Appl., 358 (2003), pp. 145–172.

[14] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[15] N. Martins, L. T. G. Lima, and H. J. C. P. Pinto, *Computing dominant poles of power system transfer functions*, IEEE Trans. Power Syst., 11 (1996), pp. 162–170.

[16] N. Martins and P. E. M. Quintão, *Computing dominant poles of power system multivariable transfer functions*, IEEE Trans. Power Syst., 18 (2003), pp. 152–159.

[17] A. M. Ostrowski, *On the convergence of the Rayleigh quotient iteration for the computation of the characteristic roots and vectors.* I, Arch. Ration. Mech. Anal., 1 (1958), pp. 233–241.

[18] A. M. Ostrowski, *On the convergence of the Rayleigh quotient iteration for the computation of the characteristic roots and vectors.* III, Arch. Ration. Mech. Anal., 3 (1959), pp. 325–340.

[19] R. D. Pantazis and D. B. Szyld, *Regions of convergence of the rayleigh quotient iteration method*, Numer. Linear Algebra Appl., 2 (1995), pp. 251–269.

[20] B. N. Parlett, *The Rayleigh quotient iteration and some generalizations for nonnormal matrices*, Math. Comp., 28 (1974), pp. 679–693.

[21] J. Rommes, *Methods for Eigenvalue Problems with Applications in Model Order Reduction*, Ph.D. thesis, Utrecht University, Utrecht, The Netherlands, 2007.

[22] J. Rommes and N. Martins, *Efficient computation of multivariable transfer function dominant poles using subspace acceleration*, IEEE Trans. Power Syst., 21 (2006), pp. 1471–1483.

[23] J. Rommes and N. Martins, *Efficient computation of transfer function dominant poles using subspace acceleration*, IEEE Trans. Power Syst., 21 (2006), pp. 1218–1226.

[24] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.

[25] J. R. Smith, J. F. Hauer, D. J. Trudnowski, F. Fatehi, and C. S. Woods, *Transfer function identification in power system application*, IEEE Trans. Power Syst., 8 (1993), pp. 1282–1290.

[26] A. Stathopoulos, *A case for a biorthogonal Jacobi–Davidson method: Restarting and correction equation*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 238–259.

[27] D. B. Szyld, *Criteria for combining inverse and Rayleigh quotient iteration*, SIAM J. Numer. Anal., 25 (1988), pp. 1369–1375.

[28] The MathWorks, Inc., *MATLAB* 7, The MathWorks, Natick, MA, 2005.

[29] A. Varga, *Enhanced modal approach for model reduction*, Math. Model Systems, 1 (1995), pp. 91–105.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 364–374

COMMENTS ON "JORDAN CANONICAL FORM OF THE GOOGLE MATRIX"*

GANG WU† AND YIMIN WEI‡

Abstract. The Google matrix is a Web hyperlink matrix which is given by $P(\alpha) = \alpha P + (1-\alpha)E$, where P is a row stochastic matrix, E is a row stochastic rank-one matrix, and $0 < \alpha < 1$. In this paper we explore the analytic expression of the Jordan canonical form and point out that a theorem due to Serra-Capizzano (cf. Theorem 2.3 in [*SIAM J. Matrix Anal. Appl.*, 27 (2005), pp. 305–312]) can be used for estimating the condition number of the PageRank vector as a function of α now viewed in the complex field. Furthermore, we give insight into a more efficient scaling matrix in order to minimize the condition number.

Key words. PageRank, Google matrix, Jordan canonical form, condition number

AMS subject classifications. 65F15, 65F10, 65C40

DOI. 10.1137/070682204

1. Preliminaries.

An important problem in Web searches is determining the importance of each page. The major ingredient in determining the order to display Web pages is PageRank [4]. The PageRank vector is the stationary distribution of the Google matrix, a stochastic and irreducible matrix whose dimension can reach 10^9 [1, 7, 11].

Analysis of the PageRank formula provides an interesting topic for the PageRank problem [6, 8, 9, 10]. Recently, Horn and Serra-Capizzano [6] and Serra-Capizzano [8, 9] determined the analytic expression of the Jordan canonical form of the Google matrix. Theorem 2.3 in [8] (see also Theorem 8.2 in [9]) is quoted as follows, which depicts the eigenvalues and invariant subspace of the Google matrix.

THEOREM 1. P n $\alpha \in (0,1)$ $E = ev^H$ n e v n $v_i \geq 0$ $||v||_1 = 1$ $P(\alpha) = \alpha P + (1-\alpha)E$ $P = XJ(1)X^{-1}$ $X = [e, x_2, \ldots, x_n]$, $Y = (X^{-1})^H = [y_1, y_2, \ldots, y_n]$,

$$J(\alpha) = \begin{bmatrix} 1 & & & & \\ & \alpha\lambda_2 & \alpha \cdot * & & \\ & & & & \\ & & & \alpha\lambda_{n-1} & \alpha \cdot * \\ & & & & \alpha\lambda_n \end{bmatrix},$$

*Received by the editors February 8, 2007; accepted for publication (in revised form) by D. P. O'Leary September 4, 2007; published electronically April 23, 2008.

http://www.siam.org/journals/simax/30-1/68220.html

†School of Mathematical Sciences, Xuzhou Normal University, Xuzhou, 221116, Jiangsu, People's Republic of China (gangwu76@yahoo.com.cn, wugangzy@gmail.com). This author was supported by the National Science Foundation of China under grant 10626044 and by the Qing-Lan Project of Jiangsu Province.

‡Corresponding author. School of Mathematical Sciences and Key Laboratory of Mathematics for Nonlinear Sciences, Fudan University, Shanghai, 200433, People's Republic of China (ymwei@fudan.edu.cn, ymwei_cn@yahoo.com). This author was supported by the National Science Foundation of China under grant 10471027 and by the Shanghai Education Committee.

(1) $$J(\alpha) = D^{-1} \begin{bmatrix} 1 & & & & \\ & \alpha\lambda_2 & * & & \\ & & \ddots & & \\ & & & \alpha\lambda_{n-1} & * \\ & & & & \alpha\lambda_n \end{bmatrix} D, \ D = \mathrm{diag}(1, \alpha, \ldots, \alpha^{n-1}),$$

(2) $$P(\alpha) = ZJ(\alpha)Z^{-1}, \quad Z = XR^{-1},$$

$$R = I + e_1 w^H, \quad w^H = (0, w_2, \ldots, w_n),$$

$$w_2 = (1-\alpha)v^H x_2/(1-\alpha\lambda_2),$$

$$w_j = \Big[(1-\alpha)v^H x_j + [J(\alpha)]_{j-1,j} w_{j-1}\Big]/(1-\alpha\lambda_j), \qquad j = 3, 4, \ldots, n.$$

We mention that in the original paper by Serra-Capizzano, there is a typo since D and D^{-1} are exchanged in (1). The following corollary due to Serra-Capizzano [8, Corollary 2.4] gives an analytic expression of the PageRank vector.

COROLLARY 2. $[y^J(\alpha)]^H$

(3) $$[y^J(\alpha)]^H = y_1^H + \sum_{j=2}^{n} w_j y_j^H.$$

As was pointed out in [8], a strong challenge posed by formula (3) is the possibility of using vector extrapolation for obtaining the expression of $[y^J(1)]^H$. The idea of the extrapolation procedure is to start from values of $[y^J(\alpha)]^H$ for some different values of α (possibly far from 1), then to compute the unknowns appearing in (3), and finally to compute $[y^J(1)]^H$. This subject is under investigation in [2, 3].

2. On the condition number of the PageRank vector. Recall from (1) and (2) that the PageRank vector $[y^J(\alpha)]^H$ is the first row of the matrix $DZ^{-1} = DRX^{-1}$, that is, $[y^J(\alpha)]^H = e_1^H(DRX^{-1})$. If we denote $W = XR^{-1}D^{-1}$, then formula (3) can be rewritten as

(4) $$[y^J(\alpha)]^H \cdot W = e_1^H.$$

It is well known that the sensitivity of the linear system (4) is closely related to the condition number $\kappa(W)$ of W [5], where

$$\begin{aligned} \kappa(W) &= ||W||\ ||W^{-1}|| \\ &= ||XR^{-1}D^{-1}||\ ||DRX^{-1}|| \\ &\le \kappa(X) \cdot \kappa(DR). \end{aligned}$$

Therefore, W will be ill-conditioned, provided either X or DR is ill-conditioned. We have the following theorem on the conditioning of DR with respect to ∞-norm.

THEOREM 3. [illegible] $D = \mathrm{diag}(1, \alpha, \ldots, \alpha^{n-1})$ [illegible] $0 < \alpha < 1$, [illegible]

$$\kappa_\infty(DR) = \max\Big\{\Big(1 + \sum_{j=2}^{n} |w_j|\Big)\Big(1 + \sum_{j=2}^{n} \frac{|w_j|}{\alpha^{j-1}}\Big),\ \frac{1}{\alpha^{n-1}}\Big(1 + \sum_{j=2}^{n} |w_j|\Big)\Big\}, \tag{5}$$

[illegible]

$$|w_2| = \frac{(1-\alpha)|v^H x_2|}{|1-\alpha\lambda_2|},$$

[illegible]

$$\begin{aligned} |w_j| \le{}& \alpha^{j-2} \cdot \frac{(1-\alpha)|v^H x_2|}{|(1-\alpha\lambda_2)\cdots(1-\alpha\lambda_j)|} + \alpha^{j-3} \cdot \frac{(1-\alpha)|v^H x_3|}{|(1-\alpha\lambda_3)\cdots(1-\alpha\lambda_j)|} \\ &+ \cdots + \frac{(1-\alpha)|v^H x_j|}{|1-\alpha\lambda_j|}, \quad j = 3, 4, \ldots, n. \end{aligned}$$

[illegible] P [illegible]

$$\kappa_\infty(DR) \ge \Big(1 + \sum_{j=2}^{n} \frac{(1-\alpha)|v^H x_j|}{|1-\alpha\lambda_j|}\Big)\ \Big(1 + \sum_{j=2}^{n} \frac{(1-\alpha)\cdot|v^H x_j|}{\alpha^{j-1}\cdot|1-\alpha\lambda_j|}\Big). \tag{6}$$

[illegible]. Since

$$R = \begin{bmatrix} 1 & w_2 & \cdots & w_n \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix},$$

it is easy to verify that

$$R^{-1} = \begin{bmatrix} 1 & -w_2 & \cdots & -w_n \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}, \qquad DR = \begin{bmatrix} 1 & w_2 & w_3 & \cdots & w_n \\ & \alpha & & & \\ & & \alpha^2 & & \\ & & & \ddots & \\ & & & & \alpha^{n-1} \end{bmatrix},$$

and

$$R^{-1}D^{-1} = \begin{bmatrix} 1 & -w_2/\alpha & -w_3/\alpha^2 & \cdots & -w_n/\alpha^{n-1} \\ & 1/\alpha & & & \\ & & 1/\alpha^2 & & \\ & & & \ddots & \\ & & & & 1/\alpha^{n-1} \end{bmatrix}.$$

Therefore,

$$||DR||_\infty = 1 + \sum_{j=2}^{n} |w_j|, \quad ||R^{-1}D^{-1}||_\infty = \max\Big\{1 + \sum_{j=2}^{n} \frac{|w_j|}{\alpha^{j-1}},\ \frac{1}{\alpha^{n-1}}\Big\},$$

and

$$\begin{aligned}\kappa_\infty(DR) &= ||DR||_\infty ||R^{-1}D^{-1}||_\infty \\ &= \max\Big\{\Big(1+\sum_{j=2}^{n}|w_j|\Big)\Big(1+\sum_{j=2}^{n}\frac{|w_j|}{\alpha^{j-1}}\Big),\ \ \frac{1}{\alpha^{n-1}}\Big(1+\sum_{j=2}^{n}|w_j|\Big)\Big\}.\end{aligned}$$

Recall from Theorem 1 that

$$w_2 = \frac{(1-\alpha)v^H x_2}{1-\alpha\lambda_2}, \qquad w_3 = \frac{(1-\alpha)v^H x_3 + (\alpha\cdot *)w_2}{1-\alpha\lambda_3},$$

where $*$ stands for either 0 or 1. Notice that

$$\begin{aligned}|w_3| &\le \frac{(1-\alpha)|v^H x_3| + \alpha\cdot|w_2|}{|1-\alpha\lambda_3|} \\ &= \alpha\cdot\frac{(1-\alpha)|v^H x_2|}{|(1-\alpha\lambda_2)(1-\alpha\lambda_3)|} + (1-\alpha)\cdot\frac{|v^H x_3|}{|1-\alpha\lambda_3|}.\end{aligned}$$

Inductively, suppose that

$$\begin{aligned}|w_{j-1}| \le \alpha^{j-3}\cdot\frac{(1-\alpha)|v^H x_2|}{|(1-\alpha\lambda_2)\cdots(1-\alpha\lambda_{j-1})|} &+ \alpha^{j-4}\cdot\frac{(1-\alpha)|v^H x_3|}{|(1-\alpha\lambda_3)\cdots(1-\alpha\lambda_{j-1})|} \\ &+\cdots+\frac{(1-\alpha)|v^H x_{j-1}|}{|1-\alpha\lambda_{j-1}|}.\end{aligned}$$

From Theorem 1, we obtain

$$w_j = \big[(1-\alpha)v^H x_j + (\alpha\cdot *)w_{j-1}\big]/(1-\alpha\lambda_j),$$

so

$$\begin{aligned}|w_j| &\le \frac{(1-\alpha)|v^H x_j| + \alpha\cdot|w_{j-1}|}{|1-\alpha\lambda_j|} \\ &\le \alpha^{j-2}\cdot\frac{(1-\alpha)|v^H x_2|}{|(1-\alpha\lambda_2)\cdots(1-\alpha\lambda_j)|} + \alpha^{j-3}\cdot\frac{(1-\alpha)|v^H x_3|}{|(1-\alpha\lambda_3)\cdots(1-\alpha\lambda_j)|} \\ &\quad +\cdots+\alpha\frac{(1-\alpha)|v^H x_{j-1}|}{|(1-\alpha\lambda_{j-1})(1-\alpha\lambda_j)|} + \frac{(1-\alpha)|v^H x_j|}{|1-\alpha\lambda_j|}, \qquad j=3,4,\ldots,n.\end{aligned}$$

Specifically, when P is diagonalizable, it follows from (5) that

$$\kappa_\infty(DR) \ge \Big(1+\sum_{j=2}^{n}|w_j|\Big)\Big(1+\sum_{j=2}^{n}\frac{|w_j|}{\alpha^{j-1}}\Big), \tag{7}$$

and recall from Theorem 2.1 in [8] that

$$|w_j| = \frac{(1-\alpha)|v^H x_j|}{|1-\alpha\lambda_j|}, \qquad j=2,3,\ldots,n, \tag{8}$$

and (6) is obtained from combining (7) and (8). □

Theorem 3 indicates that DR may be ill-conditioned as the number n in (5) and (6) is often very huge, being the total number of Web pages (in millions or billions).

Consequently, W can be ill-conditioned in practice. Actually, one is not recommended to use (3) directly. One reason is that the dimension n is huge, and the expression in (3) is simplified by replacing n with a much smaller value m [3]. Theorem 3 gives another reason: as we have just observed, W may be [illegible] even if X is [illegible] and α is far from 1, which implies that a small change in W can give a dramatic change in the PageRank vector.

However, we would like to point out that the results presented in Theorem 3 are not so strong. In Theorem 1, the matrix D is chosen as $\mathrm{diag}(1, \alpha, \ldots, \alpha^{n-1})$. In fact, the scaling matrix is not unique. For instance, if we choose $\hat{D} = \mathrm{diag}(1, \alpha^{1-n}, \ldots, \alpha^{-1})$, then $DJ(\alpha)D^{-1} = \hat{D}J(\alpha)\hat{D}^{-1}$. So it is interesting to take into account the more efficient scaling matrix D (which is not unique at all) in order to decrease the estimate of the condition number discussed in Theorem 3.

3. How to use clever choices of the scaling matrix. The conditioning for nonnegative α less than one is known to be bounded by $2/(1-\alpha)$ [7]. Therefore the interest of this paper is for α outside the unit cycle (i.e., $\alpha \in \mathcal{C}$ and $|\alpha| > 1$ [6, 9]), and to find interesting results one should use [illegible] of the scaling matrix D that minimizes the condition number. That is, we consider how to define a new matrix $\tilde{D}$ such that

$$\kappa_\infty(\tilde{D}R) = \min_{\substack{D \text{ is diagonal} \\ \text{and nonsingular}}} \kappa_\infty(DR), \tag{9}$$

with the constraint that

$$DJ(\alpha)D^{-1} = \tilde{D}J(\alpha)\tilde{D}^{-1} = \hat{J}, \tag{10}$$

where $\hat{J}$ is the Jordan canonical form of $P(\alpha)$.

However, determining an optimal matrix of minimal conditioning is a very complicated task, and the result is problem dependent. In this paper we give insight into three special cases, and the results extend easily to cover the general case, at the cost of much heavier notation. We assume from now on that $\alpha \in \mathcal{C},\ |\alpha| > 1$, and $1 - \alpha\lambda_j \neq 0$ $(j = 2, 3, \ldots, n)$ so that w_j can be well defined; see Theorem 1. Furthermore, we emphasize that all the analysis given below also applies to the case when $\alpha \in \mathcal{C},\ |\alpha| < 1$.

Let $1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of P. Then we have from Theorem 1 that the eigenvalues of $P(\alpha)$ are $1, \alpha\lambda_2, \ldots, \alpha\lambda_n$. Suppose that $|\lambda_2| = |\lambda_3| = \cdots = |\lambda_p| = 1$, and $|\lambda_j| < 1,\ j \geq p+1$. It follows from Theorem 8.2 (ii) of [6] (see also Theorem 7.2 (ii) of [9]) that $\alpha\lambda_2, \alpha\lambda_3, \ldots, \alpha\lambda_p$ are (semi)simple eigenvalues of $P(\alpha)$, thus the Jordan canonical form of $P(\alpha)$ takes the form

$$\hat{J} = [1] \oplus [\alpha\lambda_2] \oplus \cdots \oplus [\alpha\lambda_p] \oplus J_{n_1}(\alpha\nu_1) \oplus \cdots \oplus J_{n_k}(\alpha\nu_k) \tag{11}$$

and

$$J(\alpha) = [1] \oplus [\alpha\lambda_2] \oplus \cdots \oplus [\alpha\lambda_p] \oplus \alpha J_{n_1}(\nu_1) \oplus \cdots \oplus \alpha J_{n_k}(\nu_k), \tag{12}$$

where

$$J_{n_i}(\nu_i) = \begin{bmatrix} \nu_i & * & & \\ & \ddots & \ddots & \\ & & \nu_i & * \\ & & & \nu_i \end{bmatrix} \in \mathcal{C}^{n_i \times n_i}, \qquad i = 1, 2, \ldots, k,$$

and $\{\nu_1, \nu_2, \ldots, \nu_k\} \subset \{\lambda_2, \lambda_3, \ldots, \lambda_n\}$.

1. $J_{n_i}(\alpha\nu_i) = \mathrm{diag}(\alpha\nu_i, \ldots, \alpha\nu_i)$, $i = 1, 2, \ldots, k$.

In this case, $P(\alpha)$ is diagonalizable, and for any nonsingular diagonal matrix D there holds $J(\alpha) = D^{-1}\hat{J}D$. We have the following theorem.

THEOREM 4. $P(\alpha)$

$$\tilde{D} = \mathrm{diag}\Big(1,\ 1 + \sum_{j=2}^{n} |w_j|, \ldots, 1 + \sum_{j=2}^{n} |w_j|\Big), \tag{13}$$

$$\kappa_\infty(\tilde{D}R) = 1 + 2\sum_{j=2}^{n} |w_j|. \tag{14}$$

. Without loss of generality, let $D = \mathrm{diag}(1, d_2, \ldots, d_n)$ with $d_j \neq 0$, $j = 2, 3, \ldots, n$. So we have

$$DR = \begin{bmatrix} 1 & w_2 & \cdots & w_n \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix},$$

and

$$||DR||_\infty = \max\Big\{1 + \sum_{j=2}^{n} |w_j|,\ \max_{2\le j\le n} |d_j|\Big\}.$$

On the other hand,

$$R^{-1}D^{-1} = \begin{bmatrix} 1 & -w_2/d_2 & \cdots & -w_n/d_n \\ & 1/d_2 & & \\ & & \ddots & \\ & & & 1/d_n \end{bmatrix}$$

and

$$||R^{-1}D^{-1}||_\infty = \max\Big\{1 + \sum_{j=2}^{n} |w_j/d_j|,\ \frac{1}{\min_{2\le j\le n} |d_j|}\Big\}.$$

(a) If $\max_{2\le j\le n} |d_j| \le 1 + \sum_{j=2}^{n} |w_j|$, then

$$\kappa_\infty(DR) = \max\left\{\Big(1 + \sum_{j=2}^{n} |w_j|\Big)\Big(1 + \sum_{j=2}^{n} |w_j/d_j|\Big),\ \frac{1 + \sum_{j=2}^{n} |w_j|}{\min_{2\le j\le n} |d_j|}\right\}. \tag{15}$$

So as to minimize the condition number, we have to pick $\min_{2\le j\le n} |d_j|$ as large as possible. As a result, $\min_{2\le j\le n} |d_j| = \max_{2\le j\le n} |d_j| = 1 + \sum_{j=2}^{n} |w_j|$ is a reasonable choice.

(b) If $\max_{2\le j\le n} |d_j| \ge 1 + \sum_{j=2}^{n} |w_j|$, then

$$\kappa_\infty(DR) = \max\left\{\max_{2\le j\le n} |d_j|\Big(1 + \sum_{j=2}^{n} |w_j/d_j|\Big),\ \frac{\max_{2\le j\le n} |d_j|}{\min_{2\le j\le n} |d_j|}\right\}. \tag{16}$$

So as to minimize the condition number, we have to set $\max_{2\le j\le n}|d_j|$ and $\frac{\max_{2\le j\le n}|d_j|}{\min_{2\le j\le n}|d_j|}$ as small as possible. As a result, $\min_{2\le j\le n}|d_j| = \max_{2\le j\le n}|d_j| = 1+\sum_{j=2}^{n}|w_j|$ is a reasonable choice, and (14) is a direct conclusion from (15) or (16). □

2.

$$J_{n_i}(\alpha\nu_i) = \begin{bmatrix} \alpha\nu_i & 1 & & \\ & \ddots & \ddots & \\ & & \alpha\nu_i & 1 \\ & & & \alpha\nu_i \end{bmatrix} \in \mathcal{C}^{n_i\times n_i}, \qquad i = 1,2,\ldots,k.$$

In this case, all the eigenvalues $\alpha\nu_i$ $(i = 1,2,\ldots,k)$ of $P(\alpha)$ are defective, where $\{\nu_1,\nu_2,\ldots,\nu_k\} \subset \{\lambda_{p+1},\lambda_{p+2},\ldots,\lambda_n\}$, and

$$J(\alpha) = \operatorname{diag}(1,\alpha\lambda_2,\ldots,\alpha\lambda_p) \oplus \begin{bmatrix} \alpha\lambda_{p+1} & \alpha & & \\ & \ddots & \ddots & \\ & & \alpha\lambda_{n-1} & \alpha \\ & & & \alpha\lambda_n \end{bmatrix}$$

and

$$\hat{J} = \operatorname{diag}(1,\alpha\lambda_2,\ldots,\alpha\lambda_p) \oplus \begin{bmatrix} \alpha\lambda_{p+1} & 1 & & \\ & \ddots & \ddots & \\ & & \alpha\lambda_{n-1} & 1 \\ & & & \alpha\lambda_n \end{bmatrix}.$$

We consider the scaling matrix of the form $D = \operatorname{diag}(1,\ldots,1,\delta_{p+1},\ldots,\delta_n) \in \mathcal{C}^{n\times n}$. The following lemma gives a necessary and sufficient condition for the matrix D satisfying $J(\alpha) = D^{-1}\hat{J}D$.

LEMMA 5. $D = \operatorname{diag}(1,\ldots,1,\delta_{p+1},\ldots,\delta_n) \in \mathcal{C}^{n\times n}$

$$J(\alpha) = D^{-1}\hat{J}D \Longleftrightarrow \delta_j = \alpha\delta_{j-1}, \qquad j = p+2,p+3,\ldots,n. \tag{17}$$

. Note that $J(\alpha) = D^{-1}\hat{J}D \Longleftrightarrow DJ(\alpha) = \hat{J}D$. On the one hand,

$$D\cdot J(\alpha) = \operatorname{diag}(1,\alpha\lambda_2,\ldots,\alpha\lambda_p) \oplus \begin{bmatrix} \alpha\delta_{p+1}\lambda_{p+1} & \alpha\delta_{p+1} & & \\ & \ddots & \ddots & \\ & & \alpha\delta_{n-1}\lambda_{n-1} & \alpha\delta_{n-1} \\ & & & \alpha\delta_n\lambda_n \end{bmatrix}.$$

On the other hand,

$$\hat{J}\cdot D = \operatorname{diag}(1,\alpha\lambda_2,\ldots,\alpha\lambda_p) \oplus \begin{bmatrix} \alpha\delta_{p+1}\lambda_{p+1} & \delta_{p+2} & & \\ & \ddots & \ddots & \\ & & \alpha\delta_{n-1}\lambda_{n-1} & \delta_n \\ & & & \alpha\delta_n\lambda_n \end{bmatrix},$$

and (17) is obtained trivially by comparing the superdiagonal of the two matrices. □

Therefore, it follows from Lemma 5 that D takes the form

$$D = \mathrm{diag}(1, \ldots, 1, \delta_{p+1}, \alpha\delta_{p+1}, \ldots, \alpha^{n-p-1}\delta_{p+1}), \quad \delta_{p+1} \neq 0, \tag{18}$$

and the problem of defining the optimal matrix $\tilde{D}$ resorts to determining an appropriate value δ_{p+1}.

THEOREM 6. [illegible] (18). [illegible] $\alpha \in \mathcal{C}$, $|\alpha| > 1$, [illegible] $\eta = (1 + \sum_{j=2}^{n} |w_j|)/|\alpha|^{n-p-1}$ [illegible] 2 [illegible]"

[illegible]

$$\tilde{D} = \mathrm{diag}(1, \ldots, 1, \eta, \alpha\eta, \ldots, \alpha^{n-p-1}\eta), \tag{19}$$

[illegible]"

$$\kappa_\infty(\tilde{D}R) = \max\left\{\Big(1 + \sum_{j=2}^{n} |w_j|\Big)\Big(1 + \sum_{j=2}^{p} |w_j|\Big) + \sum_{j=p+1}^{n} |w_j\alpha^{n-j}|, |\alpha|^{n-p-1}\right\}. \tag{20}$$

[illegible]. For any nonsingular matrix $D = \mathrm{diag}(1, \ldots, 1, \ \delta_{p+1}, \ \alpha\delta_{p+1}, \ldots, \alpha^{n-p-1}\delta_{p+1})$, we have

$$DR = \begin{bmatrix} 1 & \cdots & w_p & w_{p+1} & \cdots & w_n \\ & \ddots & & & & \\ & & 1 & & & \\ & & & \delta_{p+1} & & \\ & & & & \ddots & \\ & & & & & \alpha^{n-p-1}\delta_{p+1} \end{bmatrix},$$

which implies

$$||DR||_\infty = \max\left\{1 + \sum_{j=2}^{n} |w_j|, \ |\alpha^{n-p-1}\delta_{p+1}|\right\},$$

since $|\alpha| > 1$. On the other hand,

$$R^{-1}D^{-1} = \begin{bmatrix} 1 & -w_2 & \cdots & -w_p & -w_{p+1}/\delta_{p+1} & \cdots & -w_n/(\alpha^{n-p-1}\delta_{p+1}) \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ & & & & 1/\delta_{p+1} & & \\ & & & & & \ddots & \\ & & & & & & 1/(\alpha^{n-p-1}\delta_{p+1}) \end{bmatrix}$$

and

$$||R^{-1}D^{-1}||_\infty = \max\left\{1 + \sum_{j=2}^{p} |w_j| + \sum_{j=p+1}^{n} \Big|\frac{w_j}{\alpha^{j-p-1}\delta_{p+1}}\Big|, \ \frac{1}{|\delta_{p+1}|}\right\}.$$

(a) If $|\delta_{p+1}| \leq (1 + \sum_{j=2}^{n} |w_j|)/|\alpha|^{n-p-1}$, then

$$\kappa_\infty(DR) = \max\left\{\Big(1 + \sum_{j=2}^{n} |w_j|\Big)\Big[1 + \sum_{j=2}^{p} |w_j| + \frac{1}{|\delta_{p+1}|}\sum_{j=p+1}^{n} \Big|\frac{w_j}{\alpha^{j-p-1}}\Big|\Big], \right.$$
$$\left. \Big(1 + \sum_{j=2}^{n} |w_j|\Big)\frac{1}{|\delta_{p+1}|}\right\}. \tag{21}$$

In order to minimize the condition number, we have to choose $|\delta_{p+1}|$ as large as possible. Since $|\delta_{p+1}| \le (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n-p-1}$, we can choose $|\delta_{p+1}| = (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n-p-1} \equiv \eta$.

(b) If $|\delta_{p+1}| \ge (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n-p-1}$, then

(22)
$$\kappa_\infty(DR) = \max\left\{|\alpha|^{n-p-1}|\delta_{p+1}|\Big(1+\sum_{j=2}^{p}|w_j|\Big)+\sum_{j=p+1}^{n}|w_j\alpha^{n-j}|,\quad |\alpha|^{n-p-1}\right\}.$$

In order to minimize the condition number, it is desirable to choose $|\delta_{p+1}|$ as small as possible. Since $|\delta_{p+1}| \ge (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n-p-1}$, we can choose $|\delta_{p+1}| = (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n-p-1} = \eta$. It is easy to see that (20) is a direct result of choosing $|\delta_{p+1}| = (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n-p-1}$ in (21) or (22). □

3.

$$J_{n_i}(\alpha\nu_i) = \begin{bmatrix} \alpha\nu_i & 0 & & & \\ & \alpha\nu_i & 1 & & \\ & & \ddots & \ddots & \\ & & & \alpha\nu_i & 1 \\ & & & & \alpha\nu_i \end{bmatrix} \in \mathcal{C}^{n_i\times n_i}, \qquad i = 1,2,\ldots,k.$$

Let $D = \operatorname{diag}(I_p, D_1, \ldots, D_k)$, where I_p is the $p\times p$ identity matrix and

$$D_i = \operatorname{diag}(d_{i_1}, d_{i_2}, \ldots, d_{i_{n_i}}) \in \mathcal{C}^{n_i\times n_i}, \qquad i = 1,2,\ldots,k,$$

are nonsingular matrices. Similar to the proof in Case 2, we have

$$D\cdot J(\alpha) = \hat{J}\cdot D \Leftrightarrow D_i\cdot \alpha J_{n_i}(\nu_i) = J_{n_i}(\alpha\nu_i)\cdot D_i \Leftrightarrow d_{i_3} = \alpha d_{i_2}, \ldots, d_{i_{n_i}} = \alpha d_{i_{n_i-1}},$$
$$i = 1,2,\ldots,k,$$

so D takes the form

(23) $$D = I_p \oplus \operatorname{diag}(1, d_{1_2}, \ldots, \alpha^{n_1-2}d_{1_{n_1}}) \oplus \cdots \oplus \operatorname{diag}(1, d_{k_2}, \ldots, \alpha^{n_k-2}d_{k_{n_k}}).$$

For simplicity, we consider D_i of the form $D_i = \operatorname{diag}(1, \delta, \alpha\delta, \ldots, \alpha^{n_i-2}\delta)$, $i = 1,2,\ldots,k$. Consequently,

(24) $$D = I_p \oplus \operatorname{diag}(1,\delta,\ldots,\alpha^{n_1-2}\delta) \oplus \cdots \oplus \operatorname{diag}(1,\delta,\ldots,\alpha^{n_k-2}\delta), \quad \delta \ne 0.$$

Partition the first row of R^{-1} conformably with D

$$[1, -w_2, \ldots, -w_n] = \big[U_p, V_1, \ldots, V_k\big],$$

where $U_p = [1, -w_2, \ldots, -w_p]$, $V_i \equiv [\tilde{w}_{i_1}, \ldots, \tilde{w}_{i_{n_i}}] \in \mathcal{C}^{n_i}$, and $\{\tilde{w}_{i_{n_1}}, \ldots, \tilde{w}_{i_{n_i}}\} \subset \{-w_{p+1}, \ldots, -w_n\}$, $i = 1,2,\ldots,k$. We have the following theorem.

THEOREM 7. [illegible] (24) [illegible] $\alpha \in \mathcal{C}$, $|\alpha| > 1$, [illegible] $n_q = \max_{1\le i\le k}\{n_i\}$ [illegible] $\mu = (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n_q-2}$ [illegible] 3 [illegible]

(25) $$\tilde{D} = I_p \oplus \operatorname{diag}(1,\mu,\alpha\mu,\ldots,\alpha^{n_1-2}\mu) \oplus \cdots \oplus \operatorname{diag}(1,\mu,\alpha\mu,\ldots,\alpha^{n_k-2}\mu),$$

(26)

$$\kappa_\infty(\tilde{D}R) = \max\left\{\left(1+\sum_{j=2}^{n}|w_j|\right)\left(1+\sum_{j=2}^{p}|w_j|+\sum_{i=1}^{k}|\tilde{w}_{i_1}|\right)+|\alpha|^{n_q-2}\Delta,\quad |\alpha|^{n_q-2}\right\},$$

$\Delta = \sum_{i=1}^{k}\left(|\tilde{w}_{i_2}|+|\tilde{w}_{i_3}|/|\alpha|+\cdots+|\tilde{w}_{i_{n_i}}|/|\alpha|^{n_i-2}\right)$

. For any matrix D that takes the form (24), we have

$$DR = \begin{bmatrix} 1 & w_2 & \cdots & w_p & \cdots & \cdots & w_n \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ & & & & D_1 & & \\ & & & & & \ddots & \\ & & & & & & D_k \end{bmatrix}$$

and

$$||DR||_\infty = \max\left\{1+\sum_{i=2}^{n}|w_i|,\ \ |\alpha|^{n_q-2}|\delta|\right\},$$

where $n_q = \max_{1\le i\le k}\{n_i\}$. It is easy to verify that

$$R^{-1}D^{-1} = \begin{bmatrix} 1 & -w_2 & \cdots & -w_p & V_1D_1^{-1} & \cdots & V_kD_k^{-1} \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ & & & & D_1^{-1} & & \\ & & & & & \ddots & \\ & & & & & & D_k^{-1} \end{bmatrix},$$

so we get

$$||R^{-1}D^{-1}||_\infty = \max\left\{||U_p||_1+\sum_{i=1}^{k}||V_iD_i^{-1}||_1,\ \frac{1}{|\delta|}\right\}.$$

Note that

$$V_iD_i^{-1} = \left[\tilde{w}_{i_1}, \tilde{w}_{i_2}/\delta, \ldots, \tilde{w}_{i_{n_i}}/(\alpha^{n_i-2}\delta)\right]$$

and

$$||V_iD_i^{-1}||_1 = |\tilde{w}_{i_1}|+\frac{1}{|\delta|}\left(|\tilde{w}_{i_2}|+|\tilde{w}_{i_3}|/|\alpha|+\cdots+|\tilde{w}_{i_{n_i}}|/|\alpha|^{n_i-2}\right),\qquad i=1,2,\ldots,k.$$

Therefore,

$$\sum_{i=1}^{k}||V_iD_i^{-1}||_1 = \sum_{i=1}^{k}|\tilde{w}_{i_1}|+\frac{1}{|\delta|}\sum_{i=1}^{k}\left(|\tilde{w}_{i_2}|+|\tilde{w}_{i_3}|/|\alpha|+\cdots+|\tilde{w}_{i_{n_i}}|/|\alpha|^{n_i-2}\right).$$

(a) If $|\delta| \le (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n_q-2}$, then

$$\kappa_\infty(DR) = \Big(1+\sum_{j=2}^{n}|w_j|\Big)\max\Big\{||U_p||_1+\sum_{i=1}^{k}||V_iD_i^{-1}||_1,\ \frac{1}{|\delta|}\Big\}. \tag{27}$$

In order to minimize the condition number, we have to choose $|\delta|$ as large as possible. Since $|\delta| \le (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n_q-2}$, we can choose $|\delta| = (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n_q-2} \equiv \mu$.

(b) If $|\delta| \ge (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n_q-2}$, then

$$\kappa_\infty(DR) = \max\Big\{|\alpha^{n_q-2}\delta|\Big(||U_p||_1+\sum_{i=1}^{k}||V_iD_i^{-1}||_1\Big),\ |\alpha|^{n_q-2}\Big\}. \tag{28}$$

In order to minimize the condition number, it is desirable to choose $|\delta|$ as small as possible. Since $|\delta| \ge (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n_q-2}$, we can choose $|\delta| = (1+\sum_{j=2}^{n}|w_j|)/|\alpha|^{n_q-2} = \mu$, and (26) is a direct conclusion from (27) or (28). □

In [9], Serra-Capizzano proposed that the "optimal" diagonal matrix of minimal conditioning can be chosen as

$$\breve{D} = I_p \oplus \mathrm{diag}(1,\alpha,\dots,\alpha^{n_1-1}) \oplus \cdots \oplus \mathrm{diag}(1,\alpha,\dots,\alpha^{n_k-1}),$$

which is obviously different from ours since the diagonal elements of the two matrices are different; see (25). Moreover, it seems that our choice is more general.

Acknowledgments. The authors would like to express their sincere thanks to Prof. R. Horn and Prof. S. Serra-Capizzano for their preprints [6, 9] and for many stimulating discussions, and to Prof. D. O'Leary for her insightful suggestions on the previous version of the manuscript.

REFERENCES

[1] Y. Bao, G. Feng, T. Y. Liu, Z. M. Ma, and Y. Wang, *Ranking websites, a probabilistic view*, Internet Math., to appear.

[2] C. Brezinski and M. Redivo-Zaglia, *Rational extrapolation for the Pagerank vector*, Math. Comput., to appear.

[3] C. Brezinski, M. Redivo-Zaglia, and S. Serra-Capizzano, *Extrapolation methods for PageRank computations*, C. R. Acad. Sci. Pairs. Ser. I, 340 (2005), pp. 393–397.

[4] S. Brin, L. Page, R. Motwami, and T. Winograd, *The PageRank Citation Ranking: Bring Order to the Web*, Technical report, Computer Science Department, Stanford University, Stanford, CA, 1998.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[6] R. Horn and S. Serra-Capizzano, *A general setting for the parametric Google matrix*, Internet Math., to appear.

[7] A. Langville and C. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ, 2006.

[8] S. Serra-Capizzano, *Jordan canonical form of the Google matrix: A potential contribution to the PageRank computation*, SIAM Matrix Anal. Appl., 27 (2005), pp. 305–312.

[9] S. Serra-Capizzano, *Google PageRanking problem: The model and the analysis*, in Proceedings of the Dagstuhl Seminar 7071, A. Frommer, M. W. Mahoney, and D. B. Szyld, eds., 2007.

[10] G. Wu, *Eigenvalues and Jordan canonical form of a successively rank-one updated complex matrix with applications to Google's PageRank problem*, J. Comput. Appl. Math., to appear.

[11] G. Wu and Y. Wei, *A Power–Arnoldi algorithm for computing PageRank*, Numer. Linear Algebra Appl., 14 (2007), pp. 521–546.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 375–396

THE METRIC NEARNESS PROBLEM*

JUSTIN BRICKELL†, INDERJIT S. DHILLON†, SUVRIT SRA†, AND JOEL A. TROPP‡

Abstract. Metric nearness refers to the problem of optimally restoring metric properties to distance measurements that happen to be nonmetric due to measurement errors or otherwise. Metric data can be important in various settings, for example, in clustering, classification, metric-based indexing, query processing, and graph theoretic approximation algorithms. This paper formulates and solves the *metric nearness problem*: Given a set of pairwise dissimilarities, find a "nearest" set of distances that satisfy the properties of a metric—principally the triangle inequality. For solving this problem, the paper develops efficient *triangle fixing* algorithms that are based on an iterative projection method. An intriguing aspect of the metric nearness problem is that a special case turns out to be equivalent to the all pairs shortest paths problem. The paper exploits this equivalence and develops a new algorithm for the latter problem using a primal-dual method. Applications to graph clustering are provided as an illustration. We include experiments that demonstrate the computational superiority of triangle fixing over general purpose convex programming software. Finally, we conclude by suggesting various useful extensions and generalizations to metric nearness.

Key words. matrix nearness problems, metric, distance matrix, metric nearness, all pairs shortest paths, triangle inequality

AMS subject classifications. 05C12, 05C85, 54E35, 65Y20, 90C06, 90C08

DOI. 10.1137/060653391

1. Introduction.

Most applications make some assumptions about the properties that the input data should satisfy. Due to measurement errors, noise, or an inability to gather data completely, an application may receive data that does not conform to its requirements. For example, imagine taking measurements as a part of some experiment. The theory suggests that the quantities measured should represent distance values amongst points in a discrete metric space. However, measurements being what they are, one ends up with a set of numbers that do not represent actual distance values, primarily because they fail to satisfy the triangle inequality. It might be beneficial to somehow optimally massage the measurements to obtain a set of "nearest" distance values that obey the properties of a metric.

It could also happen that experimental expenses and difficulties prevent one from making all the measurements. Before this incomplete set of measurements can be used in an application it might need to be tweaked, preferably minimally. As before, obtaining a "nearest" set of distance values (measurements) seems to be desirable.

Both scenarios above lead to the [illegible] : Given a set of input distances, find a "nearest" set of output distances that satisfy the properties of a metric. The notion of nearness is quantified by the function that measures distortion between the input and output distances.

Matrix nearness problems [10] offer a natural framework for pursuing the above-mentioned ideas. If there are n points, we may collect the measurements into an

*Received by the editors March 2, 2006; accepted for publication (in revised form) by R. Nabben September 11, 2007; published electronically April 23, 2008. This research was supported by NSF grant CCF-0431257, NSF Career Award ACI-0093404, and NSF-ITR award IIS-0325116. A preliminary version of this work appeared at NIPS 2004, Vancouver, Canada.

http://www.siam.org/journals/simax/30-1/65339.html

†Department of Computer Sciences, The University of Texas at Austin, Austin, TX 78712 (jlbrick@cs.utexas.edu, inderjit@cs.utexas.edu, suvrit@cs.utexas.edu).

‡Department of Mathematics, University of Michigan at Ann Arbor, Ann Arbor, MI 48109 (jtropp@umich.edu).

$n \times n$ symmetric matrix whose (j, k) entry represents the distance between points j and k. Then we seek to approximate this matrix by another (say $\boldsymbol{M}$) whose entries satisfy the triangle inequalities. That is, $m_{ij} \leq m_{ik} + m_{kj}$ for every triple (i, k, j). Any such matrix will represent the distances among n points in some metric space. We calculate approximation error with a distortion measure that depends on how the corrected matrix should relate to the input matrix. For example, one might prefer to change a few entries significantly or to change all the entries a little. This paper considers metric nearness problems that use vector norms for characterizing distortion.

There is no analytic solution to the metric nearness problem. Fortunately this problem lends itself to a convex formulation, whereby developing algorithms for solving it becomes much easier. However, despite the natural convexity of the formulations, the large number of triangle inequality constraints can make traditional approaches or general purpose convex programming software much too slow. This paper provides solutions to the metric nearness problem that exploit its inherent structure for efficiency gains.

The remainder of this paper is structured as follows. Section 1.1 highlights the principal contributions of this paper. Section 2 develops a convex formulation of the metric nearness problem. Following that, section 3 provides efficient triangle fixing algorithms for solving the metric nearness problems described in section 2. An interesting connection of metric nearness with the all pairs shortest paths (APSP) problem is studied in section 4. This connection leads to a curious new primal-dual algorithm for APSP (Algorithm 4.1).

Applications of metric nearness to clustering are discussed in section 5. Experiments highlighting running time studies and comparisons against the CPLEX software are given in section 6.1, whereas experiments illustrating the behavior of the primal-dual metric nearness algorithm are the subject of section 6.2.

Section 7.1 discusses some variations to the metric nearness problem that may also be studied. Section 7.2 describes possible future work and extensions to this paper, while two open problems are mentioned in section 7.3. Finally, section 7.4 summarizes related work and concludes this paper.

1.1. Contributions of this paper. In preliminary work [7], the authors presented the basic ideas about convex formulations of metric nearness and [illegible] algorithms. However, many of the details necessary for understanding and actually implementing the triangle fixing algorithms were missing. This paper fills that gap by presenting a detailed derivation for ℓ_1 (consequently ℓ_∞) and ℓ_2 norm–based metric nearness problems. Pseudocode for both the ℓ_1 and ℓ_2 problems is given along with the derivations.

When one allows only decreasing changes to the input, then metric nearness becomes equivalent to the APSP problem [22]. This paper studies this decrease-only version of metric nearness, and consequently obtains a new primal-dual algorithm for solving the APSP problem. This algorithm possesses some interesting characteristics related to its convergence behavior that are discussed in this paper.

The paper discusses applications to the Max-Cut problem. We also developed efficient C++ code for metric nearness that outperforms CPLEX by factors of up to 30, and it may be requested from the authors.

2. Problem formulation. We begin our formulation with a few basic definitions. We define a [illegible] to be a symmetric, nonnegative matrix with a zero diagonal. Such matrices are used to represent pairwise proximity data between objects of a certain type. A [illegible] is defined to be a dissimilarity matrix

whose entries satisfy the triangle inequalities. Specifically, $\boldsymbol{M}$ is a distance matrix if

$$m_{ij} \geq 0, \qquad m_{ii} = 0, \quad m_{ij} = m_{ji},$$

and

$$m_{ij} \leq m_{ik} + m_{kj} \quad \text{for distinct triples } (i, k, j).$$

We remark that symmetry, while part of the definition of a metric, is not crucial to our algorithms; asymmetry can be handled at the expense of doubling the running time and storage.

The distance matrices studied in this paper are assumed to arise from measuring interpoint distances between n points in a pseudometric space (i.e., two distinct points can lie at zero distance from each other). Consequently, distance matrices contain $N = \binom{n}{2}$ parameters, and we denote the set of all $n \times n$ distance matrices by $\mathscr{M}_N$. We observe that the set $\mathscr{M}_N$ is a closed, convex polyhedral cone.

Assume that the input is a dissimilarity matrix $\boldsymbol{D}$. Metric nearness seeks a distance matrix $\boldsymbol{M}$ that is closest to $\boldsymbol{D}$, with respect to some measure of "closeness." Formally, we seek a matrix $\boldsymbol{M}$ so that

$$\boldsymbol{M} \in \operatorname*{argmin}_{\boldsymbol{X} \in \mathscr{M}_N} \|\boldsymbol{X} - \boldsymbol{D}\|, \tag{2.1}$$

where $\|\cdot\|$ is a norm. Though it is possible to use any norm in the metric nearness problem (2.1), we restrict our attention to the ℓ_p norms, wherein we treat the strict upper triangular part of our matrices as vectors.

THEOREM 2.1 (attainment of minimum). [illegible] $f(\boldsymbol{X}) = \|\boldsymbol{X} - \boldsymbol{D}\|$ [illegible] $\mathscr{M}_N$ [illegible]

[illegible]. The latter claim follows immediately from the convexity of f. It remains to show that $f(\boldsymbol{X})$ always attains its minimum on the cone $\mathscr{M}_N$. For convenience, we pass to the function $g(\boldsymbol{Y}) = \|\boldsymbol{Y}\|$. Notice that if g attains a minimum on $\mathscr{M}_N - \boldsymbol{D}$, then $f(\boldsymbol{X})$ attains a minimum on $\mathscr{M}_N$. The function g is a closed convex function, and it is homogeneous of degree one, so we can compute its recession function as

$$(g0^+)(\boldsymbol{Y}) = \lim_{h \to 0} (g(h\,\boldsymbol{Y}) - g(\boldsymbol{0}))/h = \lim_{h \to 0} g(h\,\boldsymbol{Y})/h = g(\boldsymbol{Y}).$$

But g is nonnegative, so its only directions of recession are directions in which it is constant. Since $\mathscr{M}_N - \boldsymbol{D}$ is a closed, [illegible] cone, we may apply [23, Theorem 27.3] to conclude that g attains a minimum on this cone, whereby f attains its minimum on $\mathscr{M}_N$. □

2.1. Metric nearness for the ℓ_2 norm. We start with a formulation for the vector ℓ_2 norm–based metric nearness problem. Given the input dissimilarity matrix $\boldsymbol{D} = [d_{ij}]$ (where $d_{ij} = d_{ji}$), we wish to obtain a distance matrix $\boldsymbol{X}$ that minimizes the squared error

$$\frac{1}{2} \sum_{i<j} (x_{ij} - d_{ij})^2.$$

Note that the sum above ranges over $i < j$, since the involved matrices are symmetric and have a zero diagonal.

Let T_n be the set of $3\binom{n}{3}$ triples, each of which corresponds to a triangle inequality that the entries of an $n \times n$ distance matrix must satisfy. Formally,

$$T_n = \{(i,j,k),(j,k,i),(k,i,j) : 1 \le i < k < j \le n\}, \tag{2.2}$$

where the triple (i,k,j) corresponds to the triangle inequality

$$x_{ij} \le x_{ik} + x_{kj}.$$

With the introduction of an auxiliary matrix $\boldsymbol{E} = \boldsymbol{X} - \boldsymbol{D}$ that represents the changes to the original dissimilarities, the ℓ_2 metric nearness problem can be rewritten as the following quadratic program:

$$\underset{e_{ij}}{\text{minimize}} \ \frac{1}{2}\sum_{i<j} e_{ij}^2 \tag{2.3}$$

$$\text{subject to} \ \ e_{ij} - e_{ik} - e_{kj} \le d_{ik} + d_{kj} - d_{ij} = v_{ikj} \quad \text{for all } (i,k,j) \in T_n. \tag{2.4}$$

The triangle inequality constraints are encoded by (2.4). Since the ℓ_2 norm is strictly convex, the solution to (2.3) is unique. The variable v_{ikj} quantifies the violation in the (i,k,j) triangle inequality. Note that nonnegativity of x_{ij} need not be enforced explicitly as it is implied by the triangle inequalities.

2.2. Metric nearness for the ℓ_1 and ℓ_∞ norms. When measuring approximation error using the ℓ_1 norm, we wish to minimize

$$\sum_{i<j} |e_{ij}|, \tag{2.5}$$

where $e_{ij} = x_{ij} - d_{ij}$ as in the previous section. However, to write the problem as a linear program, we need to introduce additional variables $f_{ij} = |e_{ij}|$. The resulting problem is the following linear program:

$$\underset{e_{ij}, f_{ij}}{\text{minimize}} \ \ \sum_{i<j} \big(1 \cdot f_{ij} + 0 \cdot e_{ij}\big) \tag{2.6}$$

$$\begin{aligned} \text{subject to} \quad & e_{ij} - e_{ik} - e_{kj} \le v_{ikj} && \text{for all } (i,k,j) \in T_n, \\ & -e_{ij} - f_{ij} \le 0, && 1 \le i < j \le n, \\ & e_{ij} - f_{ij} \le 0, && 1 \le i < j \le n. \end{aligned} \tag{2.7}$$

The fact that $f_{ij} = |e_{ij}|$ is accomplished by the last two sets of inequalities in (2.7).

Similarly, for the ℓ_∞ nearness problem, we introduce a variable $\zeta = \max_{ij} |e_{ij}|$ that represents the vector ℓ_∞ norm of $\boldsymbol{E}$. The ℓ_∞ nearness problem becomes

$$\underset{e_{ij}, \zeta}{\text{minimize}} \ \ \zeta + \sum_{i<j} 0 \cdot e_{ij} \tag{2.8}$$

$$\begin{aligned} \text{subject to} \quad & e_{ij} - e_{ik} - e_{kj} \le v_{ikj} && \text{for all } (i,k,j) \in T_n, \\ & -e_{ij} - \zeta \le 0, && 1 \le i < j \le n, \\ & e_{ij} - \zeta \le 0, && 1 \le i < j \le n. \end{aligned} \tag{2.9}$$

The last two sets of inequalities in (2.9) express the fact $|e_{ij}| \le \zeta$ for all i and j.

2.3. Metric nearness for ℓ_p norms. Metric nearness may be easily formulated for ℓ_p norms, where $1 < p < \infty$. The problem is the following convex program:

$$\begin{aligned} &\underset{e_{ij}}{\text{minimize}} \quad \frac{1}{p}\sum_{i<j} |e_{ij}|^p \\ &\text{subject to} \quad e_{ij} - e_{ik} - e_{kj} \le v_{ikj} \quad \text{for all } (i,k,j) \in T_n. \end{aligned}$$

Since the ℓ_p norms are strictly convex for $1 < p < \infty$, the associated metric nearness problems have unique solutions. There is a basic intuition for choosing p when solving the nearness problems. The ℓ_1 norm error is computed as the absolute sum of changes to the input matrix, while ℓ_∞ reflects only the maximum absolute change. The other ℓ_p norms interpolate between these two extremes. Thus, a small value of p typically results in a solution that prefers a few large changes to the original data, while a large p typically results in a solution with many small changes. In practice, however, the ℓ_1, ℓ_2, and ℓ_∞ problems are computationally easier to solve than those using arbitrary ℓ_p norms. Thus, we focus primarily on these three problems.

3. Triangle fixing algorithms. The previous section formulated the metric nearness problem as a quadratic program for the ℓ_2 norm, as a linear program for ℓ_1 and ℓ_∞ norms, and as a convex program for ℓ_p norms. Using off-the-shelf software for these formulations might appear to be an attractive way to solve the corresponding problems. However, it turns out that the computational time and storage requirements of such an approach can be prohibitive. An efficient algorithm must exploit the inherent structure offered by the triangle inequalities. In this section, we develop [illegible], which take advantage of this structure to efficiently solve the problem for ℓ_p norms. These algorithms iterate through the triangle inequalities, optimally enforcing any inequality that is not satisfied. While enforcing the triangle inequalities, one needs to introduce appropriate correction terms to guide the iterative algorithm to the globally optimal solution. The details are provided below.

3.1. Triangle fixing for ℓ_2 metric nearness. Our approach for solving (2.3) is iterative, and is based on the technique described in [2]. Collecting all the e_{ij} values into vector $\boldsymbol{e}$ and the violation amounts v_{ijk} into $\boldsymbol{v}$, problem (2.3) may be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{e}} \quad \frac{1}{2}\boldsymbol{e}^T\boldsymbol{e} \\ &\text{subject to} \quad \boldsymbol{A}\boldsymbol{e} \le \boldsymbol{v}, \end{aligned} \tag{3.1}$$

where matrix $\boldsymbol{A}$ encodes the triangle inequalities (2.4), whereby each row of $\boldsymbol{A}$ has one $+1$ entry and two -1 entries.

The Lagrangian of (3.1) is

$$L(\boldsymbol{e}, \boldsymbol{z}) = \frac{1}{2}\boldsymbol{e}^T\boldsymbol{e} + \langle \boldsymbol{z}, \boldsymbol{A}\boldsymbol{e} - \boldsymbol{v}\rangle,$$

where $\boldsymbol{z}$ is the dual vector. A necessary condition for optimality of (3.1) is

$$\frac{\partial L}{\partial \boldsymbol{e}} = 0 \quad \Longrightarrow \quad \boldsymbol{e} = -\boldsymbol{A}^T\boldsymbol{z}, \quad \boldsymbol{z} \ge \boldsymbol{0}. \tag{3.2}$$

Using (3.2) we see that the dual problem corresponding to (3.1) is

$$\max_{\boldsymbol{z}\ge\boldsymbol{0}} \quad g(\boldsymbol{z}) = -\frac{1}{2}\boldsymbol{z}^T\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{z} - \boldsymbol{z}^T\boldsymbol{v}. \tag{3.3}$$

Algorithm 3.1. Metric nearness for ℓ_2 norm.

METRIC_NEARNESS_L2($\boldsymbol{D}$, κ)
Input: Dissimilarity matrix $\boldsymbol{D}$, tolerance κ
Output: $\boldsymbol{M} = \operatorname{argmin}_{\mathrm{X} \in \mathscr{M}_N} \|\boldsymbol{X} - \boldsymbol{D}\|_2$.
{Initialize the primal and the dual variables}
$e_{ij} \leftarrow 0$ for $1 \le i < j \le n$
$(z_{ijk}, z_{jki}, z_{kij}) \leftarrow 0$ for $1 \le i < k < j \le n$
$\delta \leftarrow 1 + \kappa$
while $(\delta > \kappa)$ {convergence test}
  foreach triangle inequality (i, k, j)
    $v \leftarrow d_{ik} + d_{kj} - d_{ij}$ {Compute violation}
    $\theta^* \leftarrow \frac{1}{3}(e_{ij} - e_{ik} - e_{kj} - v)$ $(\star)$
    $\theta \leftarrow \max\{\theta^*, -z_{ikj}\}$ {Stay within half-space of constraint}
    $e_{ij} \leftarrow e_{ij} - \theta$, $e_{ik} \leftarrow e_{ik} + \theta$, $e_{kj} \leftarrow e_{kj} + \theta$ $(\star\star)$
    $z_{ikj} \leftarrow z_{ikj} + \theta$ {Update dual variable}
  end *foreach*
  $\delta \leftarrow$ sum of changes in the e_{ij} values
end *while*
return $\boldsymbol{M} = \boldsymbol{D} + \boldsymbol{E}$

We solve (3.1) iteratively, wherein we initialize both $\boldsymbol{e}$ and $\boldsymbol{z}$ to zero as this choice satisfies (3.2). At each subsequent iteration we update the dual vector $\boldsymbol{z}$ one coordinate at a time, thereby resulting in a [illegible] procedure, while maintaining (3.2). Assume that the dual variable corresponding to inequality (i, k, j) is updated, i.e., $z'_{ikj} = z_{ikj} + \theta$. Then the corresponding update to the primal is obtained via (3.2), i.e., $\boldsymbol{e}' = \boldsymbol{e} - \theta \boldsymbol{a}_{ikj}$ (using the fact that $\boldsymbol{e}' = -\boldsymbol{A}^T \boldsymbol{z}'$), where $\boldsymbol{a}_{ikj}$ is the column vector containing the entries of the (i, j, k) [illegible] of $\boldsymbol{A}$. Recall that $\boldsymbol{a}_{ikj}$ has only three nonzero entries corresponding to the edges (i, j), (i, k), and (k, j). Thus, the update to $\boldsymbol{e}$ amounts to "fixing" (enforcing) one triangle inequality at a time, hence the name of our procedure. The parameter θ is computed by solving

$$\begin{aligned} \max_{\theta} \quad & g(\boldsymbol{z} + \theta \mathbf{1}_{ikj}) \\ \text{subject to} \quad & z_{ikj} + \theta \ge 0, \end{aligned} \tag{3.4}$$

where $\mathbf{1}_{ikj}$ indicates the standard basis vector that is zero in all positions except the ikj entry, which equals unity. Using (3.2) and (3.3), we may rewrite (3.4) as

$$\begin{aligned} \max_{\theta} \quad & g(\boldsymbol{z}) - \frac{1}{2}\|\boldsymbol{a}_{ikj}\|^2 \theta^2 + (\boldsymbol{a}_{ikj}^T \boldsymbol{e} - v_{ikj})\theta \\ \text{subject to} \quad & \theta \ge -z_{ikj}. \end{aligned} \tag{3.5}$$

Consider optimizing (3.5) in an unconstrained manner. It is easily seen that

$$\theta^* = \frac{1}{\|\boldsymbol{a}_{ikj}\|^2}(\boldsymbol{a}_{ikj}^T \boldsymbol{e} - v_{ikj}) = \frac{1}{3}(\boldsymbol{a}_{ikj}^T \boldsymbol{e} - v_{ikj}) \tag{3.6}$$

is the maximum. If $\theta^* \ge -z_{ikj}$, we are done; otherwise the maximum of (3.5) will be achieved at $\theta = -z_{ikj}$. Thus, we obtain $\theta = \max\{\theta^*, -z_{ikj}\}$ as the answer to (3.5). Algorithm 3.1 puts together all these ideas to give the complete iterative triangle fixing procedure.

The procedure derived above ensures that at each iteration $g(\boldsymbol{z}') \geq g(\boldsymbol{z})$, i.e., it is a [illegible] procedure. Following [2], it can be shown that in the limit, the $\boldsymbol{A}\boldsymbol{e} \leq \boldsymbol{v}$ constraints are satisfied. Since (3.2) is also maintained at each step, the KKT conditions, which are necessary and sufficient for this problem, are satisfied in the limit. Thus, the triangle fixing procedure converges to the optimal solution of (3.1). In fact, Algorithm 3.1 is an efficient version of Bregman's method for minimizing a convex function subject to linear inequality constraints [2]. Our algorithm exploits the structure of the problem to obtain its efficiency.

3.2. Triangle fixing for ℓ_1 and ℓ_∞. Triangle fixing is somewhat less direct for the ℓ_1 and ℓ_∞ problems. The reason these norms pose an additional challenge is because they are not strictly convex; the convergence of the basic triangle fixing procedure depends on the strict convexity of the norm used. We illustrate only the ℓ_1 case; the development for ℓ_∞ takes the same course.

With the introduction of vector and matrix notation, the ℓ_1 matrix nearness problem may be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{e},\boldsymbol{f}} \ \mathbf{0}^T\boldsymbol{e} + \mathbf{1}^T\boldsymbol{f} \\ &\text{subject to } \ \boldsymbol{A}\boldsymbol{e} \leq \boldsymbol{v}, \quad -\boldsymbol{e} - \boldsymbol{f} \leq \mathbf{0}, \quad \boldsymbol{e} - \boldsymbol{f} \leq \mathbf{0}. \end{aligned} \tag{3.7}$$

The auxiliary variable $\boldsymbol{f}$ is interpreted as the elementwise absolute value of $\boldsymbol{e}$. The violations to the triangle inequalities are again given by the vector $\boldsymbol{v}$.

To solve the linear program (3.7) without sacrificing the advantages of triangle fixing we replace it with an equivalent quadratic program. This replacement hinges upon a connection between linear and quadratic programs that may be motivated by the observation

$$\operatorname{argmin}_{\boldsymbol{g}} \|\boldsymbol{g} + \epsilon^{-1}\boldsymbol{c}\|^2 = \operatorname{argmin}_{\boldsymbol{g}} \ (\boldsymbol{g}^T\boldsymbol{g} + 2\epsilon^{-1}\boldsymbol{g}^T\boldsymbol{c} + \epsilon^{-2}\boldsymbol{c}^T\boldsymbol{c}) \approx \operatorname{argmin}_{\boldsymbol{g}} \ \boldsymbol{g}^T\boldsymbol{c}$$

if ϵ is chosen to be sufficiently small (so that the $2\epsilon^{-1}\boldsymbol{g}^T\boldsymbol{c}$ term dominates the objective function). The following theorem, which follows from a result of [17, Theorem 2.1-a-i], makes the above connection concrete.

THEOREM 3.1 (ℓ_1 metric nearness). [illegible] $\boldsymbol{g} = [\boldsymbol{e}; \boldsymbol{f}]$ [illegible] $\boldsymbol{c} = [\mathbf{0}; \mathbf{1}]$ [illegible] (3.7) [illegible] $\epsilon_0 > 0$ [illegible] $\epsilon \leq \epsilon_0$.

$$\operatorname*{argmin}_{\boldsymbol{g} \in G} \ \|\boldsymbol{g} + \epsilon^{-1}\boldsymbol{c}\|_2 = \operatorname*{argmin}_{\boldsymbol{g} \in G^\star} \ \|\boldsymbol{g}\|_2, \tag{3.8}$$

[illegible] G [illegible] (3.7) [illegible] $G^\star$ [illegible] (3.7) [illegible] (3.8).

From (3.7) one can see that the triangle inequality constraints involve only $\boldsymbol{e}$ and not $\boldsymbol{f}$. This circumstance permits us to use triangle fixing once again. As before, we go through the constraints one by one. The first $3\binom{n}{3}$ constraints are triangle constraints and are handled by triangle fixing. The remaining $2\binom{n}{2}$ absolute value constraints are very simple and thus are enforced easily.

For the ℓ_2 case, the dual variables (corresponding to each constraint) were represented by the vector $\boldsymbol{z}$. For (3.8), we let the dual variables be $[\boldsymbol{z};\ \boldsymbol{\lambda};\ \boldsymbol{\mu}]$; vector $\boldsymbol{z}$ corresponds to the triangle inequalities, while vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ correspond to $-\boldsymbol{e} - \boldsymbol{f} \leq \mathbf{0}$ and $\boldsymbol{e} - \boldsymbol{f} \leq \mathbf{0}$, respectively. Together, nonnegative values of $\boldsymbol{z}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ correspond to the feasible set G alluded to by Theorem 3.1.

Algorithm 3.2. Metric nearness for ℓ_1 norm.

METRIC_NEARNESS_L1($\boldsymbol{D}$, ϵ, κ)
Input: Dissimilarity matrix $\boldsymbol{D}$; tolerance κ; ℓ_1 parameter ϵ
Output: $\boldsymbol{M} \in \{\text{argmin}_{\mathrm{X} \in \mathscr{M}_N} \ \|\boldsymbol{X} - \boldsymbol{D}\|_1\}$
{Initialize primal and dual variables}
$e_{ij} \leftarrow 0$; $f_{ij} = -\epsilon^{-1}$ for $1 \le i < j \le n$ {Primal variables}
$(z_{ijk}, z_{jki}, z_{kij}) \leftarrow 0$ for $1 \le i < k < j \le n$ {Dual variables – triangles}
$\lambda_{ij} \leftarrow \pi_{ij} \leftarrow 0$ for $1 \le i < j \le n$ {Dual variables – Other}
$\delta \leftarrow 1 + \kappa$
while $(\delta > \kappa)$ {convergence test}
  Do triangle fixing on the e_{ij} as in Algorithm 3.1
  {Enforce $-\boldsymbol{e} - \boldsymbol{f} \le \boldsymbol{0}$ and $\boldsymbol{e} - \boldsymbol{f} \le \boldsymbol{0}$ as follows}
  $\boldsymbol{\mu} \leftarrow \frac{1}{2}(\boldsymbol{e} + \boldsymbol{f})$ {Projection parameters}
  $\boldsymbol{\theta} \leftarrow \min\{\boldsymbol{\mu}, \boldsymbol{\lambda}\}$ {Update amount}
  $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \boldsymbol{\theta}$ {Update dual vector corr. to $-\boldsymbol{e} - \boldsymbol{f} \le \boldsymbol{0}$}
  $\boldsymbol{e} \leftarrow \boldsymbol{e} - \boldsymbol{\theta}$; $\boldsymbol{f} \leftarrow \boldsymbol{f} - \boldsymbol{\theta}$ {Update primal variables}
  $\boldsymbol{\nu} \leftarrow \frac{1}{2}(\boldsymbol{f} - \boldsymbol{e})$
  $\boldsymbol{\theta} \leftarrow \min\{\boldsymbol{\nu}, \boldsymbol{\pi}\}$ {Update amount}
  $\boldsymbol{\pi} \leftarrow \boldsymbol{\pi} - \boldsymbol{\theta}$ {Update dual vector corr. to $\boldsymbol{e} - \boldsymbol{f} \le \boldsymbol{0}$}
  $\boldsymbol{e} \leftarrow \boldsymbol{e} + \boldsymbol{\theta}$; $\boldsymbol{f} \leftarrow \boldsymbol{f} - \boldsymbol{\theta}$ {Update primal variables}
  {Update convergence test parameter}
  $\delta \leftarrow$ sum of absolute changes in e_{ij}.
end.

Our augmented triangle fixing procedure is as follows. First we initialize $\boldsymbol{e}$, $\boldsymbol{f}$, $\boldsymbol{z}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ so that the first order optimality conditions derived from (3.8) are initially true. Thereafter, we enforce constraints one by one to ensure that the dual functional corresponding to (3.8) is increasing and that first order optimality conditions are maintained. Written out as Algorithm 3.2, this procedure becomes an efficient adaptation of Bregman's method, thereby, after a sufficient number of iterations, converging to the globally optimal solution.

Algorithm 3.2 depends on the parameter ϵ that governs convergence to the true optimal solution. It is an open problem to obtain an ϵ that guarantees convergence. However, upon experimentation with random dissimilarity matrices we found that setting $\epsilon^{-1} \approx \max_{ij} d_{ij}$ worked well, i.e., led to convergence, for Algorithm 3.2. Furthermore, from Theorem 3.1 we know that there exists a range within which ϵ can lie, and in practice running Algorithm 3.2 a small number (2–3) of times (with early stopping to save time) helps to determine a suitable value for ϵ for an arbitrary input matrix.

3.3. Triangle fixing for other ℓ_p norms. We can go a step further and extend triangle fixing to solve the metric nearness problem for all ℓ_p $(1 < p < \infty)$ norms. The problem may be compactly stated as

$$\min_{\boldsymbol{e}} \ \frac{1}{p} \|\boldsymbol{e}\|_p^p \qquad \text{subject to} \qquad \boldsymbol{A}\boldsymbol{e} \le \boldsymbol{v}. \tag{3.9}$$

Recall that for ℓ_2 metric nearness, at each iterative step we obtained $\boldsymbol{e}'$ from $\boldsymbol{e}$ by solving (3.2) after updating the dual variables $\boldsymbol{z}$ in a single coordinate. This update to $\boldsymbol{e}$ may be viewed as the result of an orthogonal projection of $\boldsymbol{e}$ onto the hyperplane defined by $\langle \boldsymbol{a}_{ikj}, \boldsymbol{e}' \rangle = v_{ikj}$ (ignoring inequalities for the moment). For the ℓ_p norm problem, we must instead perform a generalized projection, called a

, which involves solving the problem

$$\min_{\boldsymbol{e}'} \ \varphi(\boldsymbol{e}') - \varphi(\boldsymbol{e}) - \langle \nabla\varphi(\boldsymbol{e}), \boldsymbol{e}' - \boldsymbol{e}\rangle \quad \text{such that } \langle \boldsymbol{a}_{ikj}, \boldsymbol{e}'\rangle = \boldsymbol{v}_{ikj}, \tag{3.10}$$

where $\varphi(\boldsymbol{x}) = \frac{1}{p}\|\boldsymbol{x}\|_p^p$. We use $(\nabla\varphi(\boldsymbol{x}))_i = \operatorname{sgn}(x_i)\,|x_i|^{p-1}$ to determine the projection (3.10) by solving

$$\nabla\varphi(\boldsymbol{e}') = \nabla\varphi(\boldsymbol{e}) + \mu\boldsymbol{a}_{ikj} \qquad \text{so that} \qquad \langle \boldsymbol{a}_{ikj}, \boldsymbol{e}'\rangle = v_{ikj}. \tag{3.11}$$

Since $\boldsymbol{a}_{ikj}$ has only three nonzero entries, once again $\boldsymbol{e}$ needs to be updated in only three components. Therefore, in Algorithm 3.1 we may replace $(\star)$ by an appropriate numerical computation of the parameter μ, and replace $(\star\star)$ by the computation of the new value of $\boldsymbol{e}$ as resulting from (3.11). As before, each iteration maintains the necessary condition $\partial L(\boldsymbol{e}, \boldsymbol{z})/\partial \boldsymbol{e} = 0$ while correcting the dual vector $\boldsymbol{z}$, and the overall algorithm converges to the optimum of (3.9).

4. Metric nearness and APSP. The APSP problem [3] is an important and well-studied problem in graph theory that still continues to interest researchers. For a given weighted graph G, APSP computes an associated matrix of distances M whose entry m_{ij} gives the weight of a shortest path between vertices i and j. Optionally, shortest paths between all pairs of vertices corresponding to these distances are also obtained.

On the surface, APSP appears to have no connection with the metric nearness problem. However, it turns out that APSP can be viewed as a special case of metric nearness. We develop this connection below. Note that in the previous sections we considered only symmetric matrices. However, in this section we consider asymmetric distance matrices, which are more natural for the APSP problem, as they correspond to directed graphs.

4.1. The relation of metric nearness to APSP. Let the input be a weighted complete directed graph. We represent this graph by the (nonsymmetric) matrix $\boldsymbol{D}$, where d_{ij} denotes the edge weight of edge (i, j). On $\boldsymbol{D}$ we perform a restricted version of metric nearness that permits only decreasing changes to the d_{ij} values. Curiously this decrease-only version of metric nearness is equivalent to APSP.

LEMMA 4.1 (decrease-only metric nearness is APSP). . $\boldsymbol{M}^A \in \mathscr{M}_N$ .. . $\boldsymbol{D}$. $\boldsymbol{M}^A$ . . . . . ". . . . . $\boldsymbol{M} \in \mathscr{M}_N$ .. . . . . $\boldsymbol{D}$ . . . . $\boldsymbol{M}^A$ . . . . $\boldsymbol{M} \in \mathscr{M}_N$ .. $\boldsymbol{M} \le \boldsymbol{D}$ .. . $\boldsymbol{M} \le \boldsymbol{M}^A$

A proof of this lemma may be found in Appendix A.1. This connection between APSP and decrease-only metric nearness (DOMN) suggests that the latter may be solved by using any off-the-shelf algorithm for APSP. More interestingly, one can turn the problem around and obtain a new method to solve APSP by solving the DOMN problem. In this section, we present a new algorithm for APSP based on solving a linear programming formulation of DOMN.

APSP for dense graphs is commonly performed using the Floyd–Warshall algorithm, which has a complexity of $\Theta(n^3)$. Unlike the Floyd–Warshall algorithm, which proceeds by fixing the triangles of the graph in a predetermined order, our DOMN algorithm fixes triangles in a data-dependent order. Empirically, our algorithm converges more quickly to the solution than the Floyd–Warshall algorithm does, despite having the same asymptotic worst-case behavior.

4.2. The linear programming formulation of DOMN and its dual. Lemma 4.1 suggests that APSP solves the DOMN problem regardless of the norm used to measure the error. We, however, focus on the ℓ_1 norm problem along with its linear programming formulation. The linear program is interesting both because it is a novel formulation for solving APSP and because its dual allows us to construct shortest paths, if desired. We apply the primal-dual technique for solving the resulting linear programs and obtain a new APSP algorithm as a consequence.

4.2.1. Formulation. Let $\boldsymbol{X}$ represent a decrease-only distance matrix corresponding to the input matrix $\boldsymbol{D}$. Then the entries of $\boldsymbol{X}$ must satisfy,

$$x_{ij} \leq d_{ij} \quad \text{for all } (i,j), \tag{4.1}$$

$$x_{ij} \leq x_{ik} + x_{kj} \quad \text{for all } (i,k,j). \tag{4.2}$$

Finding the matrix with the least ℓ_1 perturbation requires solving the problem

$$\underset{x_{ij}}{\text{minimize}} \quad \sum_{ij}(d_{ij} - x_{ij}) \quad \text{subject to (4.1) and (4.2).}$$

Note that we are dealing directly with the values x_{ij} rather than the error values $e_{ij} = d_{ij} - x_{ij}$, as we did in sections 2.1 and 2.2. Since the d_{ij} are fixed we may replace this minimization by the equivalent problem

$$\begin{aligned} &\underset{x_{ij}}{\text{maximize}} \ \sum_{ij} x_{ij} \\ &\text{subject to} \quad x_{ij} \leq d_{ij} \text{ for all } (i,j), \\ &\qquad\qquad x_{ij} - x_{ik} - x_{kj} \leq 0 \quad \text{for all } (i,k,j). \end{aligned} \tag{4.3}$$

The dual problem corresponding to (4.3) is

$$\begin{aligned} &\underset{\pi_{ij}}{\text{minimize}} \ \sum_{ij} \pi_{ij} d_{ij} \\ &\text{subject to } \pi_{ij} + \sum_{k \neq i,j} (\gamma_{ikj} - \gamma_{ijk} - \gamma_{kij}) = 1 \ \text{ for all } (i,j), \\ &\qquad\qquad \pi_{ij} \geq 0 \ \text{ for all } (i,j), \\ &\qquad\qquad \gamma_{ikj} \geq 0 \ \text{ for all } (i,k,j), \end{aligned} \tag{4.4}$$

where the dual variables π_{ij} and γ_{ikj} correspond to the decrease-only constraints (4.1) and the triangle inequality constraints (4.2), respectively.

It is illustrative to cast the linear program (4.4) as a network flow problem, in which we must satisfy a demand for a single unit of flow between every pair of vertices i and j. We can accomplish this by either sending the flow [illegible] via the edge (i,j) (which corresponds to setting $\pi_{ij} = 1$) or [illegible] the edge (i,j) and routing through some other vertex k (which corresponds to setting $\gamma_{ikj} = 1$); in the latter case, we increase the demand for flow between (i,k) and (k,j) by 1.

We note that while there is a unique optimal solution to the linear program (4.3), the linear program (4.4) has several optimal solutions, some of which involve nonintegral assignments to the γ_{ikj} variables. This nonuniqueness is not unexpected, because while there is only one value that the shortest distance between two nodes in M can attain, there can be several shortest paths that achieve this distance value (paths which may contain many intermediate nodes, each of which allows a γ_{ikj} variable to assume a positive assignment).

4.3. A primal-dual algorithm for DOMN/APSP. We apply the primal-dual method [19, 16] to solve the linear programs for DOMN, and thereby obtain a new algorithm for APSP. Most treatments of the primal-dual method have a minimization of the primal problem and a maximization of the dual problem. Thus we will call (4.3) the dual problem, and (4.4) the primal problem. The primal-dual method begins with a feasible solution to the dual that is improved at each step by optimizing an [illegible] problem. In our case, we find it easier to optimize the associated restricted dual, whereby our method proceeds as follows:

1. Begin with a feasible solution to the dual problem. One such feasible solution is to set each x_{ij} to the smallest d_{ij} value.
2. Find the set P consisting of those constraints that do not have any additional slack. The decrease-only constraint $x_{ij} \leq d_{ij}$ (corresponding to dual variable π_{ij}) will be in P iff $x_{ij} = d_{ij}$, and the triangle constraint $x_{ij} - x_{ik} - x_{kj} \leq 0$ (corresponding to dual variable γ_{ikj}) will be in P iff $x_{ij} = x_{ik} + x_{kj}$.
3. Find a solution to the associated restricted dual

$$\begin{aligned} &\text{maximize} && \sum_{ij} u_{ij} \\ &\text{subject to} && u_{ij} \leq 0 && \text{if } \pi_{ij} \in P, \\ & && u_{ij} - u_{ik} - u_{kj} \leq 0 && \text{if } \gamma_{ikj} \in P, \\ & && u_{ij} \leq 1 && \text{for all } (i,j). \end{aligned} \tag{4.5}$$

   The solution u_{ij} to the associated restricted dual identifies which variables can be increased while maintaining dual feasibility.
4. If $u_{ij} = 0$, then the current value of x_{ij} is the optimal dual variable assignment. Otherwise, improve the x_{ij} assignment by adding ϵu_{ij} to x_{ij}, where ϵ is as large as possible while still maintaining dual feasibility. Return to step 2 with the new x_{ij} assignment.

By characterizing the solution of the associated restricted dual and the calculation of ϵ for the DOMN problem, we can give an efficient primal-dual algorithm. Observe that the solution to the associated restricted dual is that $u_{ij} = 1$ if the edge (i,j) is [illegible], and 0 otherwise. Computing ϵ is equivalent to determining which of the increasable edges has the least capacity for increase. Rather than use linear programming to determine the increasable edge set and computing ϵ explicitly, we can track upper bounds u_{ij} in addition to the lower bounds tracked by the x_{ij} variables. These upper bounds start as the d_{ij} values, but are reduced as edges become triangle constrained. Then the increasable set is simply the set of edges for which $x_{ij} < u_{ij}$, and ϵ is the difference between the lower bound of edges in the increasable set and the largest upper bound. Algorithm 4.1 implements these optimizations. I, the set of increasable edges, is the complement of P.

4.4. Priority queue DOMN algorithm. Algorithm 4.1 requires $O(n^4)$ time, but we can do better by noticing that the only time the lower bounds are used is to check the condition $u_e = l_e$. For edges (i,j) not in I, we have $u_{ij} = l_{ij}$, whereas all edges (i,j) in I have l_{ij} equal to the smallest upper bound. Therefore, we can replace I with a priority queue ordered by upper bound, and we do not need to keep track of lower bounds at all (even though the original dual variable values x_{ij} were lower bounds). Algorithm 4.2 implements these changes and requires only $O(n^3)$ time when implemented using a Fibonacci heap. Like the Floyd–Warshall algorithm, Algorithm 4.2 considers all edges in some order, and then fixes all triangles involving that edge. However, the Floyd–Warshall algorithm uses a fixed data-independent

Algorithm 4.1. DOMN: simple $O(n^4)$ implementation.

```
DOMN_ALG1(D)
Input: Dissimilarity matrix D
Output: M = APSP(D).
{Initialization}
u_ij ← d_ij for all i, j                          {Initial upper bounds}
x_ij ← min_{e'∈E} u_e'                            {Initial lower bounds}
I ← E                                             {Initial set of increasable edges}
while (I ≠ ∅)
    foreach (i, j) ∈ I with u_ij = x_ij
        I ← I − {(i, j)}                          {(i, j) is no longer increasable}
        foreach k ≠ i, j
            u_ik = min(u_ik, u_ij + u_jk)         {Update upper bounds}
        end foreach
    end foreach
    foreach (i, j) ∈ I
        x_ij ← min_{e'∈I} u_e'                    {Update lower bounds}
    end foreach
end while
return M where m_ij = x_ij
```

Algorithm 4.2. DOMN: improved $O(n^3)$ implementation.

```
DOMN_ALG2(D)
Input: Dissimilarity matrix D
Output: M = APSP(D).
{Initialization}
u_ij ← d_ij  for all i, j                         {Initial upper bounds}
Q.ENQUEUE((i, j), u_ij)  for all (i, j)           {Put all edges in priority-queue}
while (Q ≠ ∅)
    (i, j) ← Q.FIRST()                            {Remove edge with lowest upper bound}
    foreach k ≠ i, j
        u_ik = min(u_ik, u_ij + u_jk)             {Update upper bounds}
        Q.UPDATEPRIORITY((i, k), u_ik)            {Reorder priority queue}
    end foreach
end while
return M where m_ij = u_ij
```

order, whereas our algorithm uses a data-dependent order. As a result, our algorithm converges to the APSP/DOMN solution more rapidly, even though it still requires $O(n^3)$ time to complete.

5. An application to clustering. The metric nearness problem can be used to develop efficient algorithms for clustering that provide guarantees on the quality of the output in comparison with the optimal clustering. The MAX-CUT problem offers an especially attractive example. A cut of a graph is a partition of the vertices into two disjoint sets, and the value of a cut is the total weight of all edges that cross the partition. MAX-CUT simply asks for the cut of a graph with maximum value. If the size of each edge weight is proportional to the dissimilarity between the two vertices, solving MAX-CUT can be interpreted as finding the best clustering of the vertices into two sets.

For a general set of weights, MAX-CUT is hard enough [20] that the solution cannot be well approximated in polynomial time (unless $\mathsf{P} = \mathsf{NP}$) [1]. On the other hand, for weights that do satisfy the triangle inequality, de la Vega and Kenyon have exhibited a randomized algorithm that can approximate the solution arbitrarily well in polynomial time [5]. That is, for a given $\varepsilon > 0$, their method can (with high probability) compute in polynomial time a cut whose value is no smaller than $(1 - \varepsilon)$ times the value of the optimal cut. Of course, the time complexity grows quickly as ε shrinks.

Metric nearness plays an important role here. First, we approximate the original graph by a metric graph. Then we use the fast algorithm to produce a nearly optimal cut of the metric graph. The same cut of the original graph also has a nearly optimal value, which can be bounded in terms of the approximation error from the metric nearness problem.

THEOREM 5.1. *Let $\boldsymbol{D}$ be a dissimilarity matrix, and let $\boldsymbol{M}$ be a metric matrix. Suppose that $\mathcal{S}$ is a cut of $\boldsymbol{M}$ with value $(1-\varepsilon)\operatorname{maxcut}(\boldsymbol{M})$. Then*

$$\operatorname{cut}_{\mathcal{S}}(\boldsymbol{D}) \geq (1-\varepsilon)\operatorname{maxcut}(\boldsymbol{D}) - \left(1 - \frac{\varepsilon}{2}\right) \|\boldsymbol{M} - \boldsymbol{D}\|_1 \tag{5.1}$$

$$\operatorname{cut}_{\mathcal{S}}(\boldsymbol{D}) \geq \frac{1-\varepsilon}{\|\boldsymbol{M}/\boldsymbol{D}\|_\infty \, \|\boldsymbol{D}/\boldsymbol{M}\|_\infty} \operatorname{maxcut}(\boldsymbol{D}), \tag{5.2}$$

where $\|\cdot\|_\infty$ is the ℓ_∞ operator norm. In particular, $m_{jk} = d_{jk} = 0$ whenever (j,k) is a cut edge.

To find the optimal $\boldsymbol{M}$ for bound (5.1), we simply solve the ℓ_1 metric nearness problem. The optimal $\boldsymbol{M}$ for (5.2) cannot be obtained without solving a nonconvex optimization problem.

Proof. For a set of vertices $\mathcal{S}$, the value of the corresponding cut is computed by the linear function

$$\operatorname{cut}_{\mathcal{S}}(\boldsymbol{D}) = \sum_{j \in \mathcal{S}} \sum_{k \notin \mathcal{S}} d_{jk}.$$

The maximum cut just optimizes this functional over all subsets $\mathcal{S}$ of the vertex set $\{1, 2, \ldots, n\}$:

$$\operatorname{maxcut}(\boldsymbol{D}) = \max_{\mathcal{S}} \sum_{j \in \mathcal{S}} \sum_{k \notin \mathcal{S}} d_{jk}.$$

Obviously, $\operatorname{cut}_{\mathcal{S}}(\boldsymbol{D}) \leq \operatorname{maxcut}(\boldsymbol{D})$. It can be shown that $\operatorname{maxcut}(|\cdot|)$ is a matrix norm. In particular, it satisfies the triangle inequality for norms. It is also clear that

$$\operatorname{maxcut}(|\boldsymbol{T}|) \leq \frac{1}{2} \sum_{j \neq k} |t_{jk}| = \frac{1}{2} \|\boldsymbol{T}\|_1$$

for any symmetric matrix $\boldsymbol{T}$ with a zero diagonal.

Let us begin with bound (5.1). Suppose that $\mathcal{S}$ is a $(1-\varepsilon)$-optimal cut of $\boldsymbol{M}$.

Then

$$\begin{aligned}
\operatorname{cut}_{\mathcal{S}}(\boldsymbol{D}) &= \operatorname{cut}_{\mathcal{S}}(\boldsymbol{M}) + \operatorname{cut}_{\mathcal{S}}(\boldsymbol{D}-\boldsymbol{M}) \\
&\geq (1-\varepsilon)\ \operatorname{maxcut}(\boldsymbol{M}) - \operatorname{cut}_{\mathcal{S}}(|\boldsymbol{D}-\boldsymbol{M}|) \\
&\geq (1-\varepsilon)\ \operatorname{maxcut}(\boldsymbol{D}+(\boldsymbol{M}-\boldsymbol{D})) - \frac{1}{2}\,\|\boldsymbol{D}-\boldsymbol{M}\|_1 \\
&\geq (1-\varepsilon)\,(\operatorname{maxcut}(\boldsymbol{D}) - \operatorname{maxcut}(|\boldsymbol{M}-\boldsymbol{D}|)) - \frac{1}{2}\,\|\boldsymbol{M}-\boldsymbol{D}\|_1 \\
&\geq (1-\varepsilon)\ \operatorname{maxcut}(\boldsymbol{D}) - (1-\varepsilon/2)\,\|\boldsymbol{M}-\boldsymbol{D}\|_1.
\end{aligned}$$

The proof for the bound (5.2) follows a similar outline. First, we implicitly define a relative error matrix $\boldsymbol{E}$ with the relation $\boldsymbol{M} = \boldsymbol{D} \odot \boldsymbol{E}$. We assume that $m_{jk} = 0$ iff $d_{jk} = 0$ to ensure that $\boldsymbol{E}$ can be defined. If not, the resulting error bound would be trivial anyway. Let $r = \min\{e_{jk} : d_{jk} \neq 0\}$ and $R = \max\{e_{jk} : d_{jk} \neq 0\}$. For any zero entry of $\boldsymbol{D}$, take the corresponding entry of $\boldsymbol{E}$ in the range $[r, R]$. In what follows, we use "/" for elementwise division.

Next, observe that

$$\begin{aligned}
\operatorname{cut}_{\mathcal{S}}(\boldsymbol{M}) = \operatorname{cut}_{\mathcal{S}}(\boldsymbol{D} \odot \boldsymbol{E}) &= \sum_{j\in\mathcal{S}} \sum_{k\notin\mathcal{S}} d_{jk}\, e_{jk} \\
&\leq \max_{j\neq k} e_{jk} \sum_{j\in\mathcal{S}} \sum_{k\notin\mathcal{S}} d_{jk} \\
&\leq \|\boldsymbol{E}\|_\infty\ \operatorname{cut}_{\mathcal{S}}(\boldsymbol{D}).
\end{aligned}$$

Similarly,

$$\operatorname{maxcut}(\boldsymbol{D}) = \operatorname{maxcut}(\boldsymbol{M}/\boldsymbol{E}) \leq \|\mathbf{1}/\boldsymbol{E}\|_\infty\ \operatorname{maxcut}(\boldsymbol{M}).$$

Then we compute

$$\begin{aligned}
\operatorname{cut}_{\mathcal{S}}(\boldsymbol{D}) &\geq \frac{\operatorname{cut}_{\mathcal{S}}(\boldsymbol{M})}{\|\boldsymbol{E}\|_\infty} \\
&\geq \frac{1-\varepsilon}{\|\boldsymbol{E}\|_\infty}\ \operatorname{maxcut}(\boldsymbol{M}) \\
&\geq \frac{1-\varepsilon}{\|\boldsymbol{E}\|_\infty\,\|\mathbf{1}/\boldsymbol{E}\|_\infty}\ \operatorname{maxcut}(\boldsymbol{D}).
\end{aligned}$$

This technique can be extended to other types of problems that are computationally easier for metric graphs [12]. Mettu and Plaxton have also considered fast algorithms for clustering "nearly metric" data, but their approach relies instead on weak versions of the triangle inequality [18]. Fast approximation algorithms for various other metric problems such as k-median, MAX-TSP, etc., are discussed in [13]; our method allows extending these approximation algorithms to nonmetric data.

6. Experiments. We implemented metric nearness in C++ wherein we coded Algorithms 3.1 and 3.2. In this section we describe some experiments based on our implementation. All experiments were carried out in double precision on a P4/2.5GHz processor machine with 2GB RAM, running Linux.

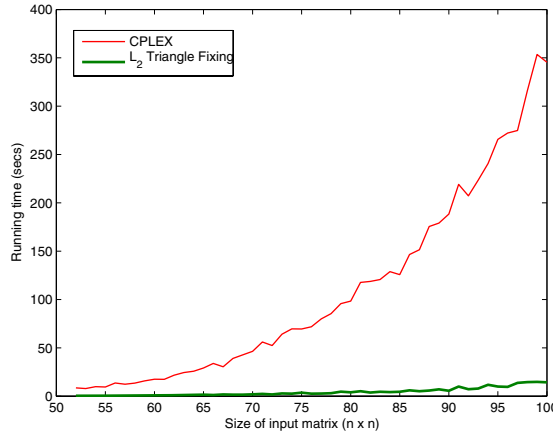


FIG. 6.1. *Running time comparison between CPLEX and the ℓ_2 triangle fixing algorithm.*

6.1. Running time. In this section we show some running time comparisons between CPLEX—a state-of-the-art linear and quadratic optimization software—and our implementations of triangle fixing. Our results clearly indicate the superiority of triangle fixing over CPLEX. For these experiments, we used random dissimilarity matrices of dimensions up to 100×100. The final values of the objective function achieved by CPLEX and our implementation agreed to five significant digits.

Figure 6.1 compares CPLEX quadratic programming to our implementation of ℓ_2 triangle fixing (see Algorithm 3.1). From the figure one can see that the triangle fixing procedure is up to 30 times faster than CPLEX's fastest method for solving the metric nearness quadratic program. Our experiments suggest that the ℓ_2 triangle fixing procedure scales as $O(n^3)$.

For ℓ_1 metric nearness, we compared CPLEX's fastest algorithm for metric nearness (determined by running all six choices available and selecting the fastest timing), and our implementation of the augmented triangle fixing procedure for solving the ℓ_1 metric nearness problem. Our implementation runs up to 15 times faster than CPLEX, as indicated by Figure 6.2. As suggested previously, we used $\epsilon = \max_{ij} d_{ij}$ for our experiments.

6.2. Decrease only metric nearness/APSP experiments. Although the Floyd–Warshall algorithm and the primal-dual algorithm, Algorithm 4.2, both have an asymptotic runtime of $O(n^3)$, the latter converges more quickly to the answer for certain classes of problems. Floyd–Warshall chooses an order of triangles to correct without any guidance, whereas the primal-dual algorithm prefers to correct triangles that include shorter edges. We can certainly imagine a problem instance where the violating triangles have longer edges, and in this case the preference for shorter edges does not help.

For randomly generated test cases, however, our primal-dual algorithm does converge more quickly than Floyd–Warshall. To illustrate this observation, we generated random matrices of dimension 200×200 that had a zero diagonal and entries between 0.1 and 10. We then determined the correct answer before running both algorithms, halting the computation at each iteration to determine the distance between the current distance matrix and the final metric. Distance was computed as the l_1 vector

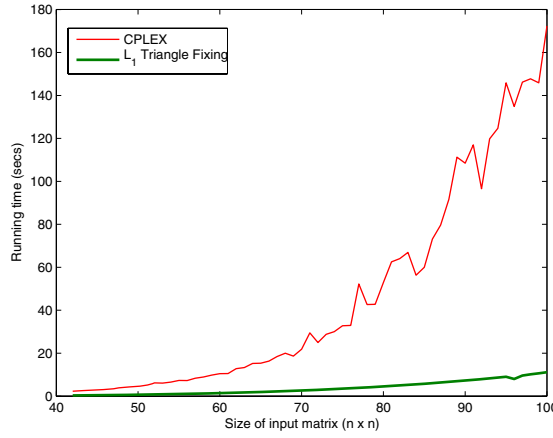


FIG. 6.2. *Running time comparison between CPLEX and augmented triangle fixing (ℓ_1).*

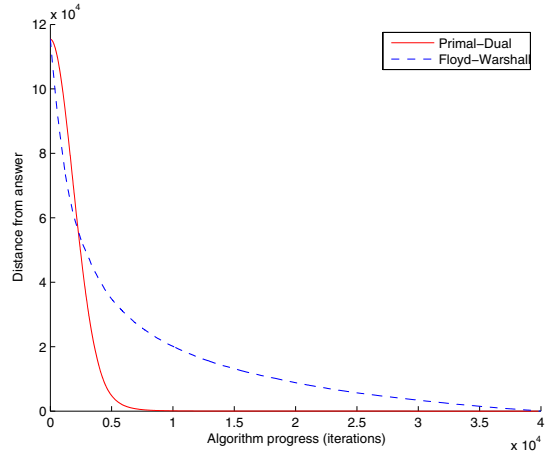


FIG. 6.3. *Convergence comparison of Floyd–Warshall and the primal-dual algorithm*

distance. Figure 6.3 gives these results, which clearly show the primal-dual algorithm converging faster than Floyd–Warshall.

To determine the approximate time to convergence as a function of n, we generated $n \times n$ matrices for values of n from 25 to 225. Figure 6.4 plots the number of iterations required to converge for both Floyd–Warshall and the primal dual algorithm. The exponents in the big-O notation runtimes were approximated by fitting the curve to the best $a \cdot n^b$ approximation. While Floyd–Warshall takes the entire $O(n^3)$ time to converge, the primal-dual algorithm converges in about $O(n^{2.8})$ time. Even more striking is Figure 6.5, which plots the number of iterations the algorithms required to converge (where nearly converging means being within $0.5 * n$ of the metric solution). Here Floyd–Warshall still required $O(n^3)$ time, but the primal-dual algorithm needed only about $O(n^{2.5})$ time.

Unfortunately, we cannot yet take advantage of this rapid convergence to improve

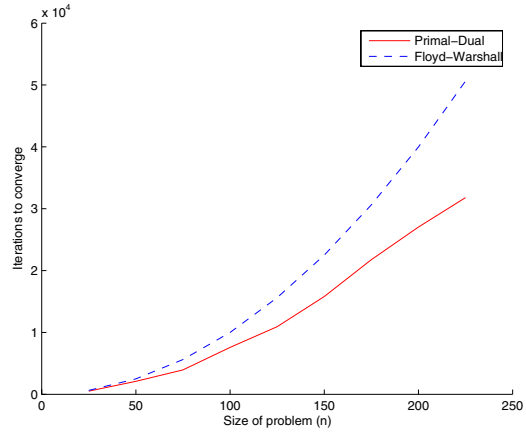


FIG. 6.4. *Iterations to converge for Floyd–Warshall and the primal-dual algorithm. Each iteration represents $O(n)$ computations.*

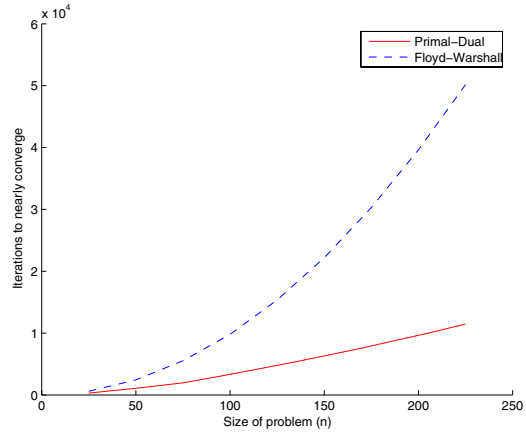


FIG. 6.5. *Iterations to nearly converge for Floyd–Warshall and the primal-dual algorithm. Each iteration represents $O(n)$ computations.*

the runtime of the APSP primal-dual algorithms. Ideally, we could terminate the algorithm when we had modified enough edges to cause the graph to be a metric. After the graph is a metric, there are no triangles in violation, so the additional steps of the algorithm do not modify the graph in any way. However, we are unaware of any computationally efficient way to solve the problem of [illegible]. That is, given a graph, return "true" if the graph is a metric, and "false" otherwise. One way to solve this is to run APSP on the graph, and then check to see if any edges were shortened. This observation yields an upper bound of $O(APSP)$ on the metricity problem. It follows that we cannot terminate the APSP primal-dual algorithms early, even if they have converged to the correct result, because testing for the termination condition has the same complexity as the problem itself.

7. Discussion. Metric nearness is a rich problem. In this paper we formally introduced the problem and derived iterative algorithms for solving it for the vector

ℓ_p norms. A special case of metric nearness was shown to be equivalent to the all pairs shortest paths problem, which led to a new algorithm for APSP. We studied applications of metric nearness to MAX-CUT clustering. Experimental results illustrate the computational advantages of triangle fixing over generic optimization methods.

7.1. Variations. One may derive numerous variations of the metric nearness problem. The simplest of these involve the modification of the triangle inequality constraints in some interesting manners. These variations are all easily solved using our framework. Examples follow.

1. In section 4 we discussed metric nearness with the restriction that permitted only decreasing changes to the entries of the input dissimilarity matrix. Similarly, one may also look at the problem where only increasing changes are permitted. Geometric or graph theoretic interpretations of this problem remain to be considered.
2. When performing metric nearness on nonsymmetric input graphs, one can choose either not to impose symmetry (as we did in the decrease-only section) or to impose symmetry. The latter case introduces additional constraints, but can be solved in our framework with only slight modifications.
3. Some applications may desire $\cdot$ or order constraints to be enforced. That is, if the input satisfies $d_{ij} < d_{pq}$, then we also require $m_{ij} < m_{pq}$. Such a requirement can be useful in scenarios where the relative ordering of the dissimilarity values has a significance for the underlying application.
4. Box constraints, i.e., constraints of type $l_{ij} \leq m_{ij} \leq u_{ij}$. Such constraints can be useful when a true metric, as opposed to a pseudometric, is desired (achieved by setting $l_{ij} > 0$). Upper bounds on the distance values may be utilized to prevent certain undesirable solutions.
5. Enforcement of λ-triangle inequalities that take the form $\lambda_{ij} m_{ij} \leq \lambda_{ik} m_{ik} + \lambda_{kj} m_{kj}$. Since the structure of the inequalities remains unaltered, this problem can also be solved by triangle fixing.

Other variations involve generalization of the basic problem. The most important of such generalizations is one that introduces a weighting scheme to the problem. Here we propose to obtain a distance matrix $\boldsymbol{M}$ such that

$$\boldsymbol{M} \in \operatorname*{argmin}_{\boldsymbol{X} \in \mathscr{M}_N} \|\boldsymbol{W} \odot (\boldsymbol{X} - \boldsymbol{D})\|,$$

where $\|\cdot\|$ is a norm, $\odot$ denotes the elementwise matrix product, and $\boldsymbol{W}$ is a weighting matrix (a symmetric nonnegative matrix). The weight matrix reflects our confidence in the entries of $\boldsymbol{D}$. When each d_{ij} represents a measurement with variance σ_{ij}^2, we might set $w_{ij} = 1/\sigma_{ij}^2$. If an entry of $\boldsymbol{D}$ is missing, one can set the corresponding weight to zero (however, the resulting problem loses strict convexity, whereby one should set this weight to a small value instead of zero).

7.2. Future work. Metric nearness is a relatively new problem. Many aspects could form a basis for future work and further consideration. The most immediate concerns that interest us are as follows:

1. Extensions to triangle fixing; for example, one may speed up the procedure by fixing all the independent triangle inequalities in parallel. One could also attempt to fix a few dependent triangle inequalities at the same time, and such an approach will result in a dual block coordinate ascent scheme [27].
2. Studying the convergence of the triangle fixing algorithms at least for the ℓ_2 case. If possible, it would be interesting to furnish a proof of convergence for

our algorithms that is independent of the convergence of the more general Bregman's method.

3. Exploring applications of metric nearness, e.g., applications to constrained clustering or various other applications that make use of proximity data and would profit from having metric data.

7.3. Open problems. Two interesting open problems spring out of metric nearness. First is the metricity problem that seeks to verify if the input dissimilarity matrix is actually a distance matrix. Some related work that probabilistically tests metric properties of an input dissimilarity matrix can be found in [21]. Whether the metric verification problem has the same complexity as the metric nearness problem remains to be ascertained. Second is the search for faster algorithms for the general metric nearness problem. Along with faster algorithms, the possibility of guaranteed polynomial-time (noniterative procedures) algorithms still remains.

7.4. Related work. Metric nearness is a relatively new problem that was introduced by the authors, where preliminary work includes [6, 7].

The most relevant research appears in recent papers of Roth et al. [24, 25]. They observe that machine learning applications often require metric data, and they propose a technique for converting general dissimilarity data into metric data. Their method, constant-shift embedding, increases all the dissimilarities by an equal amount to produce a set of Euclidean distances (i.e., a set of numbers that can be realized as the pairwise distances among an ensemble of points in a Euclidean space). The size of the translation depends on the data, so the relative and absolute changes to the dissimilarity values can be large. Our approach is completely different. We seek a consistent set of distances that [illegible] from the original measurements. In our approach, the resulting set of distances can arise from an arbitrary metric space; we do not restrict our attention to obtaining Euclidean distances. In consequence, we expect metric nearness to provide superior denoising. Moreover, our techniques can also learn distances that are missing entirely.

The technique of shifting the spectrum leads to an omission of the information carried by the negative eigenvalues of the input matrix. Laub and Müller [15] explore how the negative part of the spectrum could code for relevant features of the underlying data. Their method once again is based around computing an embedding, which is different from metric nearness, since the latter aims to only obtain a metric and constructs no embedding.

There is at least one other method for inferring a metric that proposes a technique for learning a Mahalanobis distance for data in $\mathbb{R}^s$ [28], that is, a metric $\text{dist}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^T \boldsymbol{G}(\boldsymbol{x}-\boldsymbol{y})}$, where $\boldsymbol{G}$ is an $s \times s$ positive semidefinite matrix. The user specifies that various pairs of points are similar or dissimilar. Then the matrix $\boldsymbol{G}$ is computed by minimizing the total [illegible] distances between similar points while forcing the total distances between dissimilar points to exceed one. The article provides explicit algorithms for the cases where $\boldsymbol{G}$ is diagonal and where $\boldsymbol{G}$ is an arbitrary positive semidefinite matrix. In comparison, the metric nearness problem is not restricted to Mahalanobis distances; it can learn a general discrete metric. It also allows us to use specific distance measurements and to indicate our confidence in those measurements (by means of a weight matrix), rather than forcing a binary choice of "similar" or "dissimilar."

The metric nearness problem may appear similar to metric multidimensional scaling [14], but we emphasize that the two problems are [illegible]. The latter problem endeavors to find an ensemble of points in a [illegible] metric space—usually a Eu-

clidean space—such that the distances between these points are close to the set of input distances. In contrast, metric nearness does not seek an embedding—it does not impose any hypotheses on the underlying space other than requiring it to be a metric space. For more details on Euclidean distance matrices, see [9, 10, 26].

Related to metrics are ultrametrics; a distance matrix $\boldsymbol{M}$ is said to be an ultrametric if $m_{ij} \leq \max\{m_{ik}, m_{kj}\}$ for every distinct triple of indices (i, k, j). It is known that finding the nearest (in ℓ_1 and ℓ_2 norms) ultrametric to a given input matrix is NP-complete [4]. However, the ℓ_∞-nearest ultrametric can be computed in $O(n^2)$ time [8]. Hubert, Arabie, and Meulman [11] consider the problem of representing a dissimilarity matrix by a sum of matrices having a particular form, including a form that restricts the matrices to being ultrametrics.

Appendix. More on metric nearness. This appendix includes additional informative material pertinent to metric nearness.

A.1. Metric nearness and APSP. Lemma A.1 formalizes the equivalence between a "decrease-only" version of metric nearness and APSP. This equivalence was originally suggested by [22].

LEMMA A.1 (decrease only metric nearness is APSP). . $\boldsymbol{M}^A \in \mathscr{M}_N$ . .
. $\boldsymbol{D}$ . . $\boldsymbol{M}^A$ . . . . . ."
. . . . . $\boldsymbol{M} \in \mathscr{M}_N$. $\boldsymbol{M} \leq \boldsymbol{D}$ . . $\boldsymbol{M} \leq \boldsymbol{M}^A$
. . . We prove the last statement of the lemma, noting that it immediately implies the rest.

Assume the edge weights m_{ij}^A of $\boldsymbol{M}^A$ are sorted in increasing order, and that the least-weighted edge for which $\boldsymbol{M}$ exceeds $\boldsymbol{M}^A$ is m_{ij}, i.e., $m_{ij} > m_{ij}^A$. Since $\boldsymbol{M}^A$ is an APSP solution for $\boldsymbol{D}$, each edge weight m_{ij}^A either equals d_{ij} or is the sum of weights of edges involved in a shortest path of length less than d_{ij}, as shown in Figure A.1.

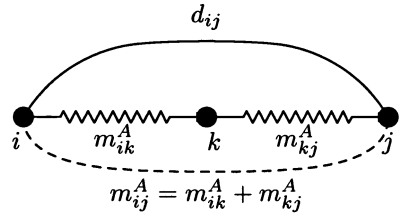


FIG. A.1. *Shortest path between i and j via k.*

In the figure, k is some intermediate vertex on a shortest path from i to j. The zig-zag lines denote paths from $i \to k$ and $k \to j$. Since $\boldsymbol{M}$ is a metric solution, $m_{ij} \leq m_{ik} + m_{kj}$. Now $m_{ik}^A \leq m_{ij}^A$ and $m_{kj}^A \leq m_{ij}^A$, since $m_{ij}^A = m_{ik}^A + m_{kj}^A$. By our assumption $m_{ij} > m_{ij}^A$ is the first place where a component of $\boldsymbol{M}$ exceeds a component of $\boldsymbol{M}^A$ (taken in sorted order), hence $m_{ik} \leq m_{ik}^A$ and $m_{kj} \leq m_{kj}^A$, which in turn implies that $m_{ij} \leq m_{ij}^A$. We have arrived at a contradiction to our initial assumption, which completes the proof of our claim. □

A.1.1. Equivalence of APSP to DOMN. Lemma A.1 shows that the optimal assignment of the x_{ij} variables in linear program (4.3) is the same as the . . . given by the APSP solution. In this section, we will investigate an equivalence between the optimal assignment of the π_{ij} and γ_{ikj} variables in linear program (4.4) and the . . . given by the APSP solution.

Given an APSP solution, we construct an optimal solution to the DOMN problem (4.3) using the following procedure:

- If the edge (i, j) is used by n shortest paths, then set $\pi_{ij} = n$.
- If the edge (i, j) is used by n shortest paths en route to node k, then set $\gamma_{ijk} = n$.

Clearly the nonnegativity constraints $\pi_{ij} \geq 0$ and $\gamma_{ikj} \geq 0$ are satisfied. We must show that the constraint

$$\pi_{ij} + \sum_{k \neq i,j} (\gamma_{ikj} - \gamma_{ijk} - \gamma_{kij}) = 1$$

is satisfied for all edges (i, j).

For vertex pairs i and j in which the shortest path from i to j is the edge (i, j), this assignment will be consistent with the constraint because all shortest paths involving the edge (i, j) fall into one of three categories: the path from i to j, paths to j that end with the edge (i, j), and paths to another vertex k that pass through the edge (i, j). The latter two categories contribute both a $+1$ and a -1 to the constraint, while the first category contributes a $+1$, resulting in a net sum of 1.

For vertex pairs i and j in which the shortest path from i to j begins with the edge (i, k), this assignment is also consistent with the constraint. There are two types of shortest paths ending at node j and using edge (i, k): the path that starts at i, and paths that start at a node l and pass through (i, k) before finishing at j. The latter type of path contributes both a $+1$ and a -1 to the constraint, while the first type contributes a $+1$ for a total of 1.

Under the proposed variable assignment procedure, the objective function for (4.4) is the sum of all path distances. Because the paths were taken from an APSP solution, this objective is minimized.

Given a optimal solution to (4.4), we construct an APSP solution using the following procedure:

- If π_{ij} is positive, then the edge (i, j) is a shortest path from i to j.
- If γ_{ikj} is positive, then there is a shortest path from i to j that passes through k; we may recursively find the shortest paths from i to k and from k to j.

REFERENCES

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and the hardness of approximation problems*, J. Assoc. Comput. Mach., 45 (1998), pp. 501–555.

[2] Y. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*, Oxford University Press, Oxford, 1997.

[3] T. H. Cormen, C. E. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed., MIT Press, Cambridge, MA, 2001.

[4] W. H. E. Day, *Computational complexity of inferring phylogenies from dissimilarity matrices*, Bull. Math. Biol., 49 (1987), pp. 461–467.

[5] W. F. de la Vega and C. Kenyon, *A randomized approximation scheme for Metric MAX-CUT*, J. Comput. System Sci., 63 (2001), pp. 531–541.

[6] I. S. Dhillon, S. Sra, and J. A. Tropp, *The Metric Nearness Problems with Applications*, Tech. report TR-03-23, Computer Sciences, University of Texas at Austin, Austin, TX, 2003.

[7] I. S. Dhillon, S. Sra, and J. A. Tropp, *Triangle fixing algorithms for the metric nearness problem*, in Advances in Neural Information Processing Systems 17, L. K. Saul, Y. Weiss, and L. Bottou, eds., MIT Press, Cambridge, MA, 2005.

[8] M. Farach, S. Kannan, and T. Warnow, *A robust model for finding optimal evolutionary trees*, Algorithmica, 13 (1995), pp. 155–179.

[9] J. C. Gower, *Properties of Euclidean and non-Euclidean distance matrices*, Linear Algebra Appl., 67 (1985), pp. 81–97.
[10] N. J. Higham, *Matrix nearness problems and applications*, in Applications of Matrix Theory, M. J. C. Gower and S. Barnett, eds., Oxford University Press, Oxford, 1989, pp. 1–27.
[11] L. J. Hubert, P. Arabie, and J. Meulman, *The representation of symmetric proximity data: Dimensions and classifications*, Computer J., 41 (1998), pp. 566–577.
[12] P. Indyk, *A sublinear-time approximation scheme for clustering in metric spaces*, in Proceedings of the 40th Annual Symposium on Foundations of Computer Science, IEEE, 1999, pp. 154–159.
[13] P. Indyk, *Sublinear time algorithms for metric space problems*, in Proceedings of the 31st Annual Symposium on Theory of Computing, ACM, 1999, pp. 428–434.
[14] J. B. Kruskal and M. Wish, *Multidimensional Scaling*, Quantitative Applications in the Social Sciences, Sage Publications, Newbury Park, CA, 1978.
[15] J. Laub and K.-R. Müller, *Feature discovery in non-metric pairwise data*, J. Mach. Learn. Res., 5 (2004), pp. 801–818.
[16] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed., Kluwer Academic, Boston, MA, 1984.
[17] O. L. Mangasarian, *Normal solutions of linear programs*, Math. Programming Stud., 22 (1984), pp. 206–216.
[18] R. R. Mettu and C. G. Plaxton, *The online median problem*, SIAM J. Comput., 32 (2003), pp. 816–832.
[19] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover, Mineola, NY, 2000.
[20] C. Papadimitriou and M. Yannakakis, *Optimization, approximation and complexity classes*, J. Comput. System Sci., 43 (1991), pp. 425–440.
[21] M. Parnas and D. Ron, *Testing metric properties*, in Proceedings of the Annual Symposium on Theory of Computing, ACM, 2001, pp. 276–285.
[22] C. G. Plaxton, *Personal communication*, 2003–2004.
[23] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[24] V. Roth, J. Laub, J. M. Buhmann, and K.-R. Müller, *Going metric: Denoising pairwise data*, in Advances in Neural Information Processing Systems 15, S. Becker, S. Thrun, and K. Obermayer, eds., MIT Press, Cambridge, MA, 2003, pp. 841–848.
[25] V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann, *Optimal cluster preserving embedding of nonmetric proximity data*, IEEE Trans. Pattern Anal. Machine Intelligence, 25 (2003), pp. 1540–1551.
[26] I. J. Schoenberg, *Remarks to Maurice Fréchet's article "Sur la definition axiomatique d'une classe d'espace distanciés vectoriellement applicable sur l'espace de Hilbert,"* Ann. of Math., 36 (1935), pp. 724–732.
[27] P. Tseng, *Dual coordinate ascent methods for nonstrictly convex minimization*, Math. Programming, 59 (1993), pp. 231–247.
[28] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, *Distance metric learning, with application to clustering with side constraints*, in Advances in Neural Information Processing Systems 15, S. Becker, S. Thrun, and K. Obermayer, eds., MIT Press, Cambridge, MA, 2003, pp. 521–528.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 397–418

COMPUTATION OF LARGE INVARIANT SUBSPACES USING POLYNOMIAL FILTERED LANCZOS ITERATIONS WITH APPLICATIONS IN DENSITY FUNCTIONAL THEORY*

C. BEKAS†, E. KOKIOPOULOU‡, AND YOUSEF SAAD§

Abstract. The most expensive part of all electronic structure calculations based on density functional theory lies in the computation of an invariant subspace associated with some of the smallest eigenvalues of a discretized Hamiltonian operator. The dimension of this subspace typically depends on the total number of valence electrons in the system, and can easily reach hundreds or even thousands when large systems with many atoms are considered. At the same time, the discretization of Hamiltonians associated with large systems yields very large matrices, whether with planewave or real-space discretizations. The combination of these two factors results in one of the most significant bottlenecks in computational materials science. In this paper we show how to efficiently compute a large invariant subspace associated with the smallest eigenvalues of a symmetric/Hermitian matrix using polynomially filtered Lanczos iterations. The proposed method does not try to extract individual eigenvalues and eigenvectors. Instead, it constructs an orthogonal basis of the invariant subspace by combining two main ingredients. The first is a filtering technique to dampen the undesirable contribution of the largest eigenvalues at each matrix-vector product in the Lanczos algorithm. This technique employs a well-selected low pass filter polynomial, obtained via a conjugate residual-type algorithm in polynomial space. The second ingredient is the Lanczos algorithm with partial reorthogonalization. Experiments are reported to illustrate the efficiency of the proposed scheme compared to state-of-the-art implicitly restarted techniques.

Key words. polynomial filtering, conjugate residual, Lanczos algorithm, density functional theory

AMS subject classifications. 65F15, 65F50

DOI. 10.1137/060675435

1. Introduction and preliminaries.

Ab initio electronic structure calculations, in the framework of density functional theory (DFT) [8, 11], have proven remarkably accurate in providing a wealth of information concerning several important physical properties of complex materials. However, DFT calculations are extremely demanding and have stretched our computational capabilities to their very limits. Therefore, advances in better simulation techniques and algorithms receive much attention in this very active field of research.

The core problem in DFT calculations is the solution of the time-independent Schrödinger equation

$$\mathcal{A}_\varrho \Psi_\varrho = E\Psi_\varrho, \tag{1}$$

*Received by the editors November 20, 2006; accepted for publication (in revised form) by D. P. O'Leary November 29, 2007; published electronically April 23, 2008. This work was supported by NSF grants NSF/ITR-0325218 and NSF/ITR-0428774, by DOE grants DE-FG02-03ER25585 and DE-FG02-03ER15491, and by the Minnesota Supercomputing Institute, and was completed while the first two authors were with the Computer Science & Engineering Department of the University of Minnesota.

http://www.siam.org/journals/simax/30-1/67543.html

†Current address: IBM, Zurich Research Laboratory, Säumerstrasse 4, CH-8803, Rüschlikon, Switzerland (bek@zurich.ibm.com).

‡Current address: EPFL, Signal Processing Institute - ITS, CH-1015, Lausanne, Switzerland (effrosyni.kokiopoulou@epfl.ch).

§Computer Science & Engineering Department, University of Minnesota, Twin Cities, 200 Union St. SE, Minneapolis, MN 55455 (saad@cs.umn.edu).

where ϱ is the charge density of the electrons distribution, $\mathcal{A}_\varrho$ is the Hamiltonian operator, Ψ_ϱ are the wavefunctions, and E is the energy of the system. Observe that this is a nonlinear eigenvalue problem, since the Hamiltonian and the wavefunctions depend upon each other through the charge density ϱ. Recent decades have seen many methods that attempt to efficiently solve (1). All of them utilize some sort of iteration which aims to improve some initially selected wavefunctions so that at the end of the iteration the approximate energy E is as small as possible, or in other words the solution of (1) is self-consistent.

The charge density $\varrho(r)$ at a point r in space is calculated from the eigenfunctions Ψ_i of the Hamiltonian $\mathcal{A}$ via the formula

$$\varrho(r) = \sum_{i=1}^{n_o} |\Psi_i(r)|^2, \tag{2}$$

where the summation is taken over all occupied states (valence electrons) n_o of the system under study. This is a crucial calculation in DFT since the potential V of the Hamiltonian $\mathcal{A} = \nabla^2 + V$ depends on the charge density ϱ, which in turn depends on the eigenvectors Ψ_i of $\mathcal{A}$ (see (2)), and, as a result, an iterative loop is required to achieve self-consistence. Computing the charge density $\varrho(r)$ via (2) requires eigenvectors, though it is more accurate to say that what is needed is an orthogonal basis of the invariant subspace associated with the n_o algebraically smallest eigenvalues of the Hamiltonian. This is because $\varrho(r)$ is invariant under orthogonal transformations of the basis of eigenfunctions $\{\Psi_i\}$. If the symmetric matrix A is the discretization of the Hamiltonian $\mathcal{A}$ and the vectors ψ_i are the corresponding discretizations of the eigenfunctions $\Psi_i(r)$ with respect to r, then the charge densities are the diagonal entries of the "functional density matrix"

$$P = Q_{n_o} Q_{n_o}^\top \quad \text{with} \quad Q_{n_o} = [\psi_1, \ldots, \psi_{n_o}]. \tag{3}$$

Specifically, the charge density at the jth point r_j is the jth diagonal entry of P. In fact, any orthogonal basis $\mathcal{Q}$ which spans the same subspace as the eigenvectors ψ_i, $i = 1, \ldots, n_o$, can be used. This observation has led to improved schemes which do not focus on extracting individual eigenvectors. For example, [1] showed that the semiorthogonal basis computed by the Lanczos algorithm with partial reorthogonalization can be used in order to extract accurate approximations to the charge density. This scheme results in substantial savings relative to schemes which rely on the full reorthogonalization of the Lanczos vectors and the accurate calculations of the eigenvectors. When using standard diagonalization software, much attention is paid to obtaining accurate eigenvectors, at a cost that is often quite high. If one focuses on invariant subspaces, all that is needed is that a good basis of the subspace be computed, but this basis does not need to be a basis of accurate eigenvectors. For example, a set of m vectors which are linearly independent and which are known to have no components in the undesired eigenvectors will constitute such a basis, and an orthonormal basis can be obtained from it if we want to compute the charge density ϱ. Approximate eigenvectors can be extracted from this basis (by a Rayleigh–Ritz projection process) but this is not necessary. Shifting the focus from individual eigenvectors to bases of invariant subspaces can reduce the cost considerably.

In simple terms, the problem considered in this paper can be stated as follows. Given a real symmetric (or complex Hermitian) matrix $A \in \mathbb{R}^{n \times n}$ with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, compute the invariant subspace $\mathcal{S}_{n_o}$ associated with the

eigenvalues which do not exceed a certain limit γ. In electronic structures, γ is the Fermi energy level and the interval $[a, \gamma]$ contains the (algebraically) smallest occupied eigenstates $\lambda_1, \ldots, \lambda_{n_o}$. We assume that we are given an interval $[\alpha, \beta]$ which (tightly) contains the spectrum of A. The nature of the algorithms used in this paper also requires that $\alpha \geq 0$. If this is not satisfied, we shift matrix A by a scalar σ so that $A + \sigma I$ does not have any negative eigenvalues. Methods for computing an interval $[\alpha, \beta]$ when this is not readily available are discussed in section 3.1.

When the number of desired eigenvalues is rather small, say on the order of a few dozen, the problem can be addressed by a number of successful algorithms. Among these is an extensively used general purpose method based on implicitly restarted Lanczos iterations [33], and implemented in the software package `ARPACK` [16]. However, the problem becomes much harder in the case when we seek to compute the invariant subspace associated with a large number of eigenvalues that reach deep into the interior of the spectrum of the matrix at hand. Indeed, in electronic structure calculations, the dimension of the corresponding invariant subspace is equal to the number of occupied states n_o, which typically depends upon the number of free electrons of the system under study. Current state-of-the-art calculations may involve hundreds or even thousands of states. In addition, the dimension n of the Hamiltonian A also depends on the number of atoms and the topology of the system and is typically on the order of a few hundred thousand to several million.

The method proposed in this paper exploits two distinct and complementary tools to address the problem stated above. The first is a filtering technique which is used to dampen the undesirable contribution of the largest eigenvalues at each matrix-vector product in the Lanczos algorithm. This technique employs a well-selected low pass filter polynomial, obtained via a conjugate residual- (CR-)type algorithm in polynomial space. The second ingredient is the Lanczos algorithm with partial reorthogonalization. The main rationale for this approach is that filtering will help reduce the size of the Krylov subspace required for convergence, and this will result in substantial savings both in memory and in computational costs.

Earlier papers presented these two tools in the literature. For example, the filter polynomial used here is borrowed from [28], and earlier variants were used in [12] and [6]. The use of the partial reorthogonalization Lanczos (PR-Lanczos) was suggested in [1]. However, one of the difficulties with the method in [1] is that very large bases are often required. Thus, the goal of the present paper is to show how to effectively combine these two distinct and powerful tools, namely, polynomial filtering on the one hand and PR-Lanczos on the other, to solve the difficult problem of extracting large invariant subspaces. The motivation for using polynomial filtering in various applications, including computing large invariant subspaces, was also discussed in [28].

1.1. Previous work. An alternative viewpoint which appears in existing DFT codes is to replace diagonalization by "direct minimization," which in effect amounts to computing the subspace of minimum trace, i.e., an orthogonal basis $Q = [q_1, \ldots, q_{n_0}]$ such that $\mathrm{tr}(Q^T A Q)$ is minimum. In fact, many publications of the mid 1990s focused on avoiding orthogonality, which turned out to be hard to achieve. A method that was explicitly based on "trace-minimization" was proposed by Sameh and Wisniewski [30] as far back as 1982. Many methods used in planewave codes are variants of the same theme and are similar to subspace iteration and trace-min iteration. They begin with a certain subspace of size n_o (or close) and then improve each vector individually while the others are fixed. Clearly, when iterating on the ith vector, orthogonality

must be enforced against the first $i-1$ vectors. While this does not refer directly to eigenvectors, the algorithm implicitly computes these eigenvectors individually.

Other codes offered an alternative to this type of scheme in the form of the block-Davidson algorithm. When planewave bases are used, it is easy to precondition the eigenvalue problem for a number of reasons [25]. For example, preconditioners for eigenvalue problems in which planewaves are used can be easily extracted from matrices which use lower-dimensional representations of the Hamiltonian, i.e., Hamiltonians obtained from using fewer planewaves and extended to higher dimensions in some simple way. The lower-dimensional Hamiltonians have good representatives of the desired eigenvectors of the higher-dimensional ones, and this class of preconditioners can be quite effective in this context. In real-space methods, the situation is quite different. In this case, we found that preconditioning the eigenvalue problem is much harder [29]. Generally the gains with the standard preconditioners that were attempted were small, and these are outweighed by the additional cost of applying the preconditioner and by the loss of the 3-term recurrence of the Lanczos procedure. Specifically, one can potentially use the Lanczos procedure with an inexpensive form of reorthogonalization, but this is no longer possible with the Davidson approach, which requires a full orthogonalization at each step. In [1] we explored this approach. The Lanczos algorithm was adapted in a number of ways, the most important of which was to replace the reorthogonalization step by a partial reorthogonalization scheme [15, 24, 31, 32].

The use of matrix polynomials and filtering has been used in other ways, and the idea has played a prominent role in linear scaling and related methods; see, for example, [9, 10, 17, 21, 22]. In some cases, these methods will consist of computing the entire density matrix (3) [21] or a small part of it as an approximation [10].

1.2. The Lanczos procedure. The Lanczos algorithm [14] (see also [3, 4, 7, 24, 26]) builds a sequence of vectors $q_1, q_2, \ldots, q_m$ which form an orthonormal basis $Q_m \in \mathbb{R}^{n\times m}$ of the Krylov subspace

$$\mathcal{K}_m(A, q_1) = \mathtt{span}\{q_1, Aq_1, A^2q_1, \ldots, A^{m-1}q_1\}, \tag{4}$$

where q_1 is an arbitrary (typically random) initial vector with $\|q_1\| = 1$. As is well known, this sequence of vectors satisfies the 3-term recurrence

$$\beta_{i+1}q_{i+1} = Aq_i - \alpha_i q_i - \beta_i q_{i-1}. \tag{5}$$

Note that each step of the Lanczos algorithm requires the matrix A only in the form of matrix-vector products, which can be quite appealing in some situations, such as when A is available in stencil form.

If $Q_m = [q_1, \ldots, q_m]$ and if T_m denotes the symmetric tridiagonal matrix

$$T_m = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{bmatrix}, \tag{6}$$

where the scalars α_i, β_i are computed by the Lanczos algorithm, then it can be verified that $AQ_m = Q_mT_m + \beta_{m+1}q_{m+1}e_m^\top$, where e_m is the mth column of the canonical basis and q_{m+1} is the last vector computed by the Lanczos algorithm.

The eigenvalues of matrix A are approximated by those of matrix T_m. The Lanczos algorithm quickly yields good approximations to extremal eigenvalues of A. In contrast, convergence is typically much slower for the interior of the spectrum [24].

2. Computing large eigenspaces with the Lanczos procedure. In the situation when large eigenspaces are to be computed by the Lanczos algorithm, the number m of Lanczos steps required for all the desired eigenvectors to converge can be quite large. Therefore, if the algorithm is to be applied without any form of restarting or preconditioning, then we will have to deal with two related demands: (1) the need to apply some form of reorthogonalization to the Lanczos vectors [1, 15, 24, 31, 32], and (2) the need to store the Lanczos basis Q_m because it is needed by the reorthogonalization steps. The first constraint increases computational cost, and some care must be exercised for the reorthogonalization process not to become too expensive. The second raises the issue of memory costs. Storing the Lanczos basis Q_m will require a large memory size, and may even force one to resort to secondary storage.

Note that reorthogonalization will ultimately require all basis vectors to be fetched from main memory and that the cost of orthogonalizing the vector q_k against all previous ones will incur a cost of $O(kn)$, which yields a quadratic total cost of $O(m^2n)$ when summed over m steps. This cost will eventually overwhelm any other computation done, and it is the main reason why so many attempts have been made in the past to avoid or reduce the orthogonalization penalty in electronic structures codes; see, e.g., [5, 13, 18, 19, 36].

Note also that there is an additional severe penalty due to memory traffic as the size of the system increases, because modern processors work at a much faster rate than memory subsystems. It was argued in [1] that memory requirements do not necessarily pose a significant problem for the matrix sizes encountered and the machines typically in use for large calculations. For example, storing 2000 vectors of length 1 million requires "only" 16 GB of memory, which is certainly within reach of most high-performance computers.[1] However, for larger calculations this will be an enormous burden and out-of-core algorithms would be needed.

2.1. Use of partial reorthogonalization. A remarkable property of the Lanczos algorithm is that, in theory (exact arithmetic), it computes a basis of the Krylov subspace, which is , . This is done with a simple 3-term recurrence. However, in practice, i.e., in the presence of finite precision arithmetic (e.g., double precision floating point arithmetic), the basis vectors quickly start to lose orthogonality. The onset of loss of orthogonality is sudden and takes place as soon as one or more eigenvectors start converging, as was discovered in the seminal work of Paige [23]. As soon as this happens, the orthogonality is completely lost very rapidly, indicating an unstable underlying computation. As an illustration, consider the Hamiltonian ($n = 17077$) corresponding to $\mathtt{Si}_{10}\mathtt{H}_{16}$, which was obtained by the real-space code PARSEC.[2] We test the orthogonality of the bases Q_i, $i = 1, \ldots, m$, with $m = 200$ by computing the norm $\|Q_i^\top Q_i - I_i\|_2$, where I_i is the identity matrix of size i. The left plot in Figure 1 illustrates the rapid deterioration of orthogonality among basis vectors.

A number of existing reorthogonalization schemes are often employed to remedy the problem. The simplest of these consists of a full reorthogonalization approach,

[1] In modern high-performance computers this will typically be available in a single node.
[2] http://www.ices.utexas.edu/parsec/index.html

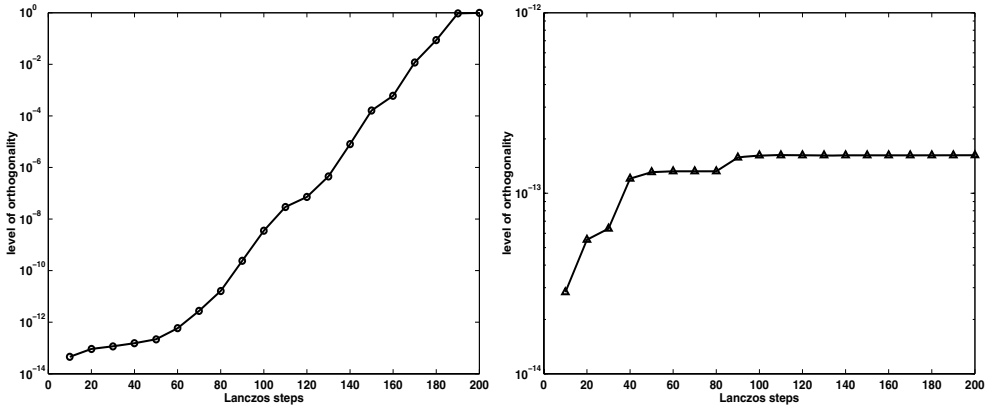


FIG. 1. *Levels of orthogonality of the Lanczos basis for the Hamiltonian ($n = 17077$) corresponding to* $Si_{10}H_{16}$. *Left: Lanczos without reorthogonalization. Right: Lanczos with partial reorthogonalization. The number of reorthogonalizations was* 34 *with an additional* 3400 *inner vector products.*

whereby the orthogonality of the basis vector q_i is enforced against all previous vectors at each step i. This means that the vector q_i, which in theory is already orthogonal against $q_1, \ldots, q_{i-1}$, is orthogonalized (a second time) against these vectors. In principle, we no longer have a 3-term recurrence, but this is not an issue as the corrections are small and usually ignored (see, however, Stewart [34]). However, full reorthogonalization can be a costly procedure.

An alternative is [illegible], which attempts to reorthogonalize only when it is deemed necessary. The goal is not so much to guarantee that the vectors are exactly orthogonal as to ensure that they are at least nearly orthogonal. Typically, the loss of orthogonality is allowed to grow to roughly the square root of the machine precision before a reorthogonalization is performed. A result by Simon [31] ensures that we can get fully accurate approximations to the Ritz values (eigenvalues of the tridiagonal matrix T_m) in spite of a reduced level of orthogonality among the Lanczos basis vectors. Furthermore, a key to the successful utilization of this result is the existence of clever recurrences which allow us to estimate the level of orthogonality among the basis vectors [15, 32]. It must be stressed that the cost of updating the recurrence is very modest. Let $\omega_{i,j} = q_i^\top q_j$ denote the "loss of orthogonality" between any basis vectors q_i and q_j. Then the following is the so-called ω-recurrence [32]:

$$\beta_i \omega_{i+1,j} = (\alpha_j - \alpha_i)\omega_{i,j} + \beta_{j-1}\omega_{i,j-1} - \beta_{i-1}\omega_{i-1,j}, \tag{7}$$

where the scalars α_i and β_i, $i = 1, \ldots,$ are identical to the ones computed by the Lanczos algorithm.

Thus, we can cheaply and efficiently probe the level of orthogonality of the current vector (say q_i) and determine whether a reorthogonalization step against previous basis vectors is required. The right plot in Figure 1 illustrates the corresponding level of orthogonality when partial reorthogonalization is applied. Only 34 reorthogonalization steps were required, compared with the 200 that would have been required if full reorthogonalization was employed.

It was shown in [1] that partially reorthogonalized Lanczos combined with techniques that avoid explicit computation of eigenvectors can lead to significant savings

in computing charge densities for electronic structure calculations. Partial reorthogonalization will play a key role in the algorithm to be described in the next section.

2.2. Polynomial acceleration and restarting techniques. The above discussion strongly suggests that it is critical to use a Lanczos basis that is as small as possible. In order to achieve this, we can apply the Lanczos process not to the original matrix A but rather on a matrix $p(A)$, where $p(t)$ is a polynomial of small degree, designed to be close to zero for large eigenvalues and close to one for the eigenvalues of interest. Of course, polynomial acceleration in Krylov techniques is not a new idea (see, for example, [26] and the references therein). Typically, the goal is to restart the Lanczos procedure after a fixed number of iterations with a starting vector from which unwanted eigendirections have been filtered out. In this paper we follow a different approach. We do not employ any restarts, but rather filter each matrix-vector product in the Lanczos process using a small number of CR-type iterations with the matrix A. As can be expected, the proposed scheme will require a much smaller number of basis vectors than without filtering. However, each matrix-vector product is now more costly. Experiments will show that the trade-off is in favor of filtering.

Observe that in exact arithmetic, if the starting vector q_1 is orthogonal to an eigenvector ψ_j, then the Krylov subspace $\mathcal{K}_m$ will never have any components in ψ_j, regardless of the number of steps m. Restarting techniques utilize this property to speed up the computation of the desired invariant subspace. The goal is to progressively construct a starting vector $q_1^{(k)}$, which at each restart k will have larger components in desired eigendirections, and smaller ones in undesired eigendirections. In contrast to the standard Lanczos procedure, the dimension of the Krylov subspace is not allowed to grow indefinitely. When a maximum number of iterations $M_{\max}$ is reached, a new starting vector $q_1^{(k+1)}$ is selected and the process is restarted; see [26, 33] for details.

Whether explicit or implicit, restarting can be designed to filter out eigendirections corresponding to eigenvalues $\lambda_j > \lambda_{n_o}$. The goal is to accelerate convergence towards the algebraically smallest eigenvalues. However, round-off will cause eigendirections in the largest eigenvalues to quickly reappear. This is illustrated in Figure 2. The matrix that is tested corresponds to a second order finite difference approximation of the two-dimensional Laplace differential operator. The starting vector is the sum

$$q_1 = \sum_{k=1}^{n_o} \psi_i$$

of the eigenvectors corresponding to the smallest $n_o = 200$ eigenvalues of the matrix. The left plot of Figure 2 illustrates that at the first step of the Lanczos procedure, the vector q_1 is orthogonal (up to machine precision) to the unwanted eigenvectors. However, it takes only $m = 13$ steps of Lanczos for the coefficients in the largest eigenvectors to start dominating the last basis vector q_m.

What happened can be easily explained. Let ϵ denote the machine precision and assume that $\langle q_1, \psi_i \rangle = \epsilon$ for a given eigenvector ψ_i with $i > n_o$. Recall that the Lanczos vector q_{m+1} is of the form $q_{m+1} = z_m(A)q_1$, where z_m is a polynomial of degree m, called the $(m+1)$st Lanczos polynomial. The sequence of polynomials z_k, $k = 1, \ldots, m$, is orthogonal with respect to a certain discrete inner product. Since the initial vector has very small components in the eigenvectors associated with eigenvalues $\lambda_i > \lambda_{n_o}$, it is to be expected that the Lanczos polynomial z_m is such

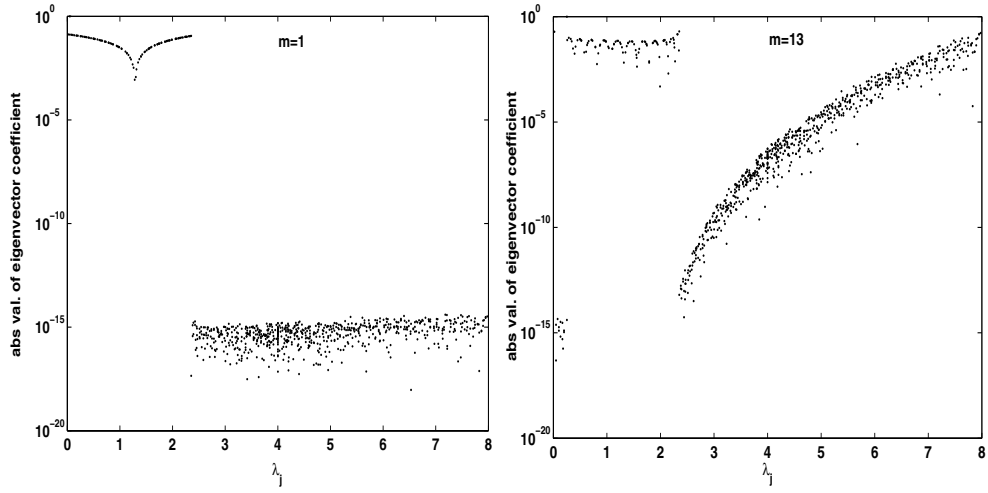


FIG. 2. *Coefficients of the last basis vector q_m of the Lanczos procedure (no partial reorthogonalization was required) for the discretization of the Laplacian, when the starting vector does not have any components in undesired eigenvectors. Left: one step. Right: $m = 13$ steps.*

that $z_m(\lambda_i) \gg 1$ for $i > n_o$. Therefore, we will have

$$\begin{aligned} \langle q_{m+1}, \psi_i \rangle &= \langle z_m(A) q_1, \psi_i \rangle \\ &= \langle q_1, z_m(A) \psi_i \rangle \\ &= z_m(\lambda_i) \langle q_1, \psi_i \rangle \\ &= z_m(\lambda_i) \epsilon. \end{aligned} \tag{8}$$

As a result, the small component ϵ will be amplified by the factor $z_m(\lambda_i)$, which is likely to be very large.

The situation can be remedied by replacing A by an operator of the form $B = p(A)$, where $p(\lambda_i)$ is small. If B is used in the Lanczos algorithm, then note that every time we multiply q by B, a component in the direction ψ_i that is small (relative to the others) will remain small.

Before we state the result in detail, we must recall that in inexact arithmetic, the Lanczos relation (5) is replaced by a relation of the form

$$A q_i = \beta_{i+1} q_{i+1} + \alpha_i q_i + \beta_i q_{i-1} - z_i, \tag{9}$$

where z_i is an error vector which, in general, remains small.

LEMMA 2.1. $\lambda > \lambda_{n_o}$ ψ $\delta \equiv p(\lambda)$ $\{q_i\}$ (9) $\epsilon_i^{\psi} = \langle \psi, z_i \rangle$ $\sigma_i = \langle q_i, \psi \rangle$

$$\beta_{i+1}\sigma_{i+1} + (\alpha_i - \delta)\sigma_i + \beta_i \sigma_{i-1} = \epsilon_i^{\psi} \tag{10}$$

$\beta_{m+1} e_1^\top (T_m - \delta I)^{-1} e_m \neq 0$ σ_{m+1} q_{m+1} ψ

$$\sigma_{m+1} = \frac{\varepsilon_m^\top (T_m - \delta I)^{-1} e_1 - \sigma_1}{\beta_{m+1} e_m^\top (T_m - \delta I)^{-1} e_1}, \tag{11}$$

[illegible] $\varepsilon_m = [\epsilon_1^\psi, \epsilon_2^\psi, \ldots, \epsilon_m^\psi]^\top$ [illegible] T_m [illegible] (6)
[illegible]. Let $B \equiv p(A)$. We begin with the relation

$$Bq_i = \beta_{i+1} q_{i+1} + \alpha_i q_i + \beta_i q_{i-1} - z_i.$$

Taking the inner product with ψ yields

$$\langle Bq_i, \psi \rangle = \beta_{i+1} \langle q_{i+1}, \psi \rangle + \alpha_i \langle q_i, \psi \rangle + \beta_i \langle q_{i-1}, \psi \rangle - \epsilon_i^\psi.$$

Since $B\psi = \delta\psi$, this readily yields the expression (10).

Define the vector $s_m = [\sigma_1, \sigma_2, \ldots, \sigma_m]^\top$. We can rewrite the relations (10) for $i = 1, \ldots, m$ in matrix form as

$$(T_m - \delta I) s_m = \varepsilon_m - \beta_{m+1} \sigma_{m+1} e_m,$$

which yields the relation $s_m = (T_m - \delta I)^{-1} \varepsilon_m - \beta_{m+1} \sigma_{m+1} (T_m - \delta I)^{-1} e_m$. Now, we add the condition that σ_1 is known:

$$\sigma_1 = e_1^\top s_m = e_1^\top (T_m - \delta I)^{-1} \varepsilon_m - \beta_{m+1} \sigma_{m+1} e_1^\top (T_m - \delta I)^{-1} e_m,$$

from which we obtain the desired expression (11). □

The main point of the above lemma is that it explicitly provides the amplification factor for the coefficient in the direction ψ in terms of computed quantities. This factor is the denominator of the expression (11). Note that in exact arithmetic, the vector ε_m is zero and the initial error of σ_1 in the direction of ψ is divided by the factor $\beta_{m+1} e_1^\top (T_m - \delta I)^{-1} e_m$. We can obtain a slightly simpler expression by "folding" the term σ_1 into the vector ε_m. This is helpful if σ_1 is of the same order as the ϵ_i^ψ's as it simplifies the expression. Set

$$\hat{\varepsilon}_m = \varepsilon_m - \sigma_1 (T_m - \delta I) e_1.$$

Note that only ϵ_1^ψ and ϵ_2^ψ are modified into $\hat{\epsilon}_1^\psi = \epsilon_1^\psi - (\alpha_1 - \delta)\sigma_1$ and $\hat{\epsilon}_2^\psi = \epsilon_2^\psi - \beta_2 \sigma_1$, while the other terms remain unchanged, i.e., $\hat{\epsilon}_i^\psi = \epsilon_i^\psi$ for $i > 2$. Then (11) becomes

$$\sigma_{m+1} = \frac{\hat{\varepsilon}_m^\top (T_m - \delta I)^{-1} e_1}{\beta_{m+1} e_m^\top (T_m - \delta I)^{-1} e_1}. \tag{12}$$

Let us consider the unfavorable scenario first. When $B \equiv A$ then T_m is simply the tridiagonal matrix obtained from the Lanczos algorithm and δ is an eigenvalue of A. Assume that $\lambda = \lambda_n$, the largest (unwanted) eigenvalue. Even if q_1 has very small components in the direction of λ, convergence will eventually take place (see (8)), and T_m will tend to have an eigenvalue close to λ, so $(T_m - \delta I)^{-1} e_1 \equiv y_m$ is close to an eigenvector of T_m associated with its largest eigenvalue. As is well known, the last components of (converged) eigenvectors of T_m will tend to be much smaller than the first ones. Therefore, if $\hat{\varepsilon}_m$ is a small random vector, then σ_m will become larger and larger because the numerator will converge to a certain quantity while the denominator will converge to zero.

The use of a proper inner polynomial $p(t)$ prevents this from happening early by [illegible]. In this situation δ is an eigenvalue of B among many others that are clustered around zero, so convergence is considerably slower towards the corresponding eigenvector. By the time convergence takes place, the desirable subspace will have already been computed.

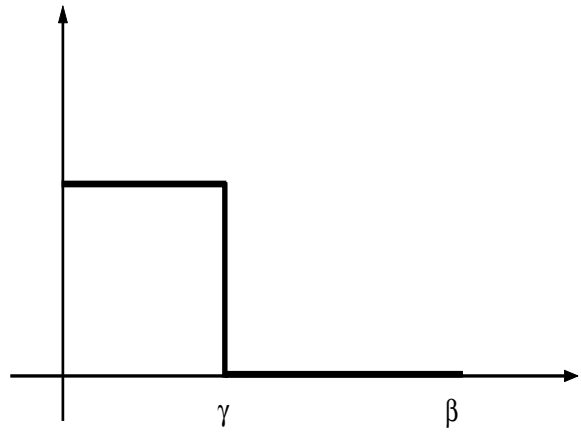


FIG. 3. *The Heaviside function for the interval* $[\gamma, \beta]$.

3. The filtered Lanczos procedure. Partial reorthogonalization can significantly extend the applicability of Lanczos in electronic structure calculations (see [1]), but there are computational issues related to the use of very long Lanczos bases when a large invariant subspace is sought. These issues can be addressed by employing polynomial filtering in the Lanczos procedure.

In exact arithmetic, the ideal solution to this problem is to use an initial vector which is filtered so that it has no eigencomponents associated with λ_i, $i > n_o$. However, we saw earlier that in the course of the Lanczos procedure, components along the largest eigenvectors will quickly return. We discussed the reasons for this behavior and suggested a simple remedy which consists of replacing the matrix-vector product Aq_i in the usual Lanczos algorithm by $p(A)q_i$, where $p(t)$ is a low-degree polynomial filter that approximates the Heaviside function (see Figure 3). The interval $[\gamma, \beta]$ contains all the unwanted (largest) eigenvalues, which are approximately mapped by $p(t)$ to zero.

All that is required to implement the proposed filtered Lanczos scheme is to substitute the matrix-vector product Aq_i with a function $\mathcal{P}(A, q_i, d)$ which evaluates the product of the matrix polynomial $p(A)$ with the vector q_i. Let d be the degree of the polynomial $p(t)$. Then the cost per step of the filtered Lanczos procedure, compared with the plain Lanczos procedure, is d additional matrix-vector products.

Observe that the filtered Lanczos process constructs an approximate invariant subspace for the matrix $p(A)$ which is also an invariant subspace for A itself. However, while the restriction of $p(A)$ on the orthogonal Lanczos basis Q_m is a tridiagonal matrix, i.e., $Q_m^\top p(A) Q_m = T_m$ is tridiagonal, this is no longer true for A, i.e.,

$$Q_m^\top A Q_m = \tilde{T}_m, \tag{13}$$

where $\tilde{T}_m$ is in general dense. The eigenvalues of A are approximated by those of $\tilde{T}_m$, while the eigenvalues of T_m approximate those of $p(A)$. However, A and $p(A)$ have the same eigenvectors. Thus, if we consider the matrix of normalized eigenvectors Y of T_m and $\tilde{Y}$ of $\tilde{T}_m$, respectively, then approximations to the eigenvectors of A are given either by the columns of the matrices $Q_m Y$ or $Q_m \tilde{Y}$. Furthermore, approximations to the eigenvalues of A are available from the eigenvalues of $\hat{T}_m = Y^\top Q_m^\top A Q_m Y$.

Similarly to the Lanczos procedure, the basis vectors q_i in the filtered Lanczos

procedure are also expected to rapidly lose orthogonality. Thus, the partial reorthogonalization techniques of section 2.1 will prove to be particularly useful in the practical deployment of the method.

The larger the degree of the polynomial $p(t)$, the closer it can be made to the Heaviside function. On the other hand, using a larger degree d will induce a higher computational cost. It is important to note that in practice we do not seek to approximate the Heaviside function everywhere on its domain of definition. We would like the polynomial $p(t)$ to take small values on the region of the unwanted eigenvalues. Section 4 discusses a CR-type iteration that achieves this goal. In order to describe the filtered Lanczos iteration, it suffices for the time being to consider the application of the filtering polynomial as a "black box" function $\mathcal{P}(A, q_i, d)$.

3.1. The algorithm. In order to compute a basis for an invariant subspace $\mathcal{S}_{n_o}$ for the n_o algebraically smallest eigenvalues of matrix A, we assume that we are given an interval $(\gamma, \beta]$, which contains all the unwanted eigenvalues $\lambda_j > \lambda_{n_o}$. Assuming that the matrix A does not have any negative eigenvalues, it suffices to consider only the left endpoint γ of the interval. In electronic structure calculations, the problem is often a variation of this one, in that we wish to compute an invariant subspace associated with the n_o smallest eigenvalues. However, there is an outer loop, and previous information can be used to obtain a good interval on which to restrict the search. There are also instances where the number of eigenvalues n_o is unknown, but rather we are given an upper bound γ for the eigenvalues that need to be considered.

Starting vector. It is important that the starting vector q_1 be free of components in the undesired eigenvectors. To this end we apply a high-degree polynomial filter p_h on a random vector $\tilde{q}$, such that $q_1 = p_h(A)\tilde{q}$. The degree of this first polynomial can be quite high (say $d_h = 200$ or so) to get a good elimination of the undesired components. A systematic way to stop this initial iteration is to monitor the norm of the residual of the CR iteration, which indicates how well the sequence of the orthogonal CR polynomials approximate the base filter function. Once this norm, which is cheap to compute (see section 4.1), falls below a user-specified tolerance, the iteration is stopped. This in turn will guarantee that unwanted large eigendirections have been adequately dampened.

Bounding intervals. If we are not given an interval $[\alpha, \beta]$ that tightly contains the eigenvalues, then we employ a number of unrestarted Lanczos iterations in order to obtain approximations for the bounds α and β. In practice, the number of these iterations is kept low. Let r_1 and r_n be the residual vectors for the approximate extremal eigenvalues $\tilde{\lambda}_1$ and $\tilde{\lambda}_n$ of matrix A obtained from a few Lanczos steps. Then we use the practical bounds $\tilde{\alpha} = \tilde{\lambda}_1 - \|r_1\|$ and $\tilde{\beta} = \tilde{\lambda}_n + \|r_n\|$. If $\tilde{\alpha}$ is negative, then we shift the Hamiltonian so as to make all its eigenvalues positive. The "interval of wanted eigenvalues" is user-defined. Standard "self-consistent field" iterations in electronic structure methods are inherently nonlinear iterations and as such information from previous iterations can be exploited to obtain a good estimate for the wanted interval. At the beginning of the self-consistent field loop, there are adequate initializations based on superposition of atomic wavefunctions that can be exploited. The "unwanted" interval is readily defined from the above.

Inner polynomial transformation. The main Lanczos iteration will be performed with a filter polynomial of A, i.e., the Lanczos algorithm is run with $B = p(A)$. The degree d of p is much smaller than that of p_h, in order to reduce the overall cost. Typically $d \equiv 8$.

Convergence criterion. Equally important in limiting the computational cost is the convergence test. Let $(\tilde{\lambda}_i, \tilde{x}_i)$ be an approximate eigenpair, where $x_i = Q_m y_i$ and $(\tilde{\lambda}_i, y_i)$ is an eigenpair of the dense matrix $\tilde{T}_m$ (13). Then it is natural to monitor the norm of the residual $r_i = A\tilde{x}_i - \tilde{\lambda}_i\tilde{x}_i$. It is well known (see, e.g., [24]) that

$$\|r_i\| = \|A\tilde{x}_i - \tilde{\lambda}_i\tilde{x}_i\| = |\beta_{m+1}|\,|y_i^m|,$$

where y_i^m is the last element of the eigenvector y_i. In order to reduce the computational cost as well as the memory load, we opt to avoid calculating the eigenvectors y_i at every step. Thus, we choose to monitor, during the iteration, the sum of the eigenvalues $\tilde{\lambda}_i$ of matrix $\tilde{T}_k$, which correspond to those eigenvalues of A that are smaller than the upper bound γ, $s_k = \sum_{\tilde{\lambda}_i<\gamma} \tilde{\lambda}_i$. Only when the change in the sum s_k, in comparison to s_{k-1}, is less than a user-defined tolerance do we calculate the eigenvectors y_i and thus check convergence by means of the residual norms $\|r_i\|$. If all residual norms are adequately small, then we stop the iteration. Otherwise, we continue the iteration and repeat the process. Further savings are achieved by not performing the convergence test for s_k at every Lanczos step, but only infrequently, for example, at fixed intervals.

Computation of the projection matrix $\tilde{T}_m$. Observe that

$$\tilde{T}_i = Q_{i+1}^\top A\, Q_{i+1} = [Q_i \quad q_{i+1}]^\top A[Q_i \quad q_{i+1}] = \begin{bmatrix} Q_i^\top AQ_i & Q_i^\top Aq_{i+1} \\ q_{i+1}^\top AQ_i & q_{i+1}^\top Aq_{i+1} \end{bmatrix}. \tag{14}$$

Thus, matrix $\tilde{T}_m$ can be computed incrementally during the course of the algorithm. Obviously, if $\tilde{T}_m$ is updated at every step i, then no additional memory is required. However, a more efficient `BLAS 3` implementation is possible if we postpone the update of $\tilde{T}_m$ and rather perform it at fixed intervals (which can be made to coincide with the intervals at which convergence is checked). This will come at the expense of a few additional vectors in memory. In particular, we will have to store the vectors Aq_{i+1} for a number of consecutive steps.

Figure 4 shows a high-level algorithmic description of the filtered Lanczos iteration.

4. Polynomial filters. This section focuses on the problem of defining and applying the polynomial filter. Details on the algorithms described here can be found in [28]. We begin with a brief summary of filtering techniques when solving linear systems of equation by "regularization" [20]. In regularized solution methods, one seeks to find an approximate solution to the linear system $Ax = b$ by inverting A only in the space associated with the largest eigenvalues, leaving the other part untouched. As was explained in [28], computing a filtered solution amounts to computing a vector $s(A)b$ whose residual vector $p(A)b = b - As(A)b$ is a certain filter polynomial, typically one that is computed to be close to 1 for small eigenvalues and close to 0 for larger eigenvalues. In other words, it would resemble the desired filter polynomial, such as the one shown on the right of Figure 7.

The approximate solutions produced by Krylov subspace methods for solving a linear system $Ax = b$ are of the form $s_j(A)r_0$, where s_j is a polynomial of degree $\leq j$. The corresponding residual vector is $p_{j+1}(\lambda) = 1 - \lambda s_j(\lambda)$. This polynomial is of degree $j + 1$ and has value 1 at $\lambda = 0$. In standard (unfiltered) methods one attempts to make the polynomial $\lambda s_j(\lambda)$ close to the function 1 on the (discrete) set of eigenvalues. Chebyshev methods attempt to make the polynomial $\lambda s(\lambda)$ close to the function 1,

FILTERED LANCZOS ALGORITHM.
(,,. . *)*
Matrix $A \in \mathbb{R}^{n\times n}$, starting vector q_1, $\|q_1\|_2 = 1$,
polynomial filter function $\mathcal{P}(A, q, d)$ that approximates the step function,
high polynomial degree d_h, stride `strd`, upper bound γ

(. .,. . *)*
Eigenvalues of A smaller than γ and orthogonal basis $Q = [q_1, q_2, \ldots]$ for the invariant subspace associated with these eigenvalues

1. Set $\beta_1 = 0$, $q_0 = 0$
2. Thoroughly filter initial vector $q_1 = \mathcal{P}(A, q_1, d_h)$, $q_1 = q_1/\|q_1\|$
3. **for** $i = 1, \ldots$
4. $w_i = \mathcal{P}(A, q_i, d_l) - \beta_i q_{i-1}$
5. $\alpha_i = \langle w_i, q_i \rangle$
6. $w_i = w_i - \alpha_i q_i$
7. $\beta_{i+1} = \|w_i\|_2$
8. **if** $(\beta_{i+1} == 0)$ **then** `stop`
9. $q_{i+1} = w_i/\beta_{i+1}$
10. **if** `rem`$(i, \ldots)$ $== 0$ **then**
11. Compute last row/column of matrix $\tilde{T}_i = Q_i^\top A Q_i$
11. Compute all eigenvalues $\tilde{\lambda}_j$ of $\tilde{T}_i$ such that $\tilde{\lambda}_j < \gamma$
12. Compute $s_i = \sum_{\tilde{\lambda}_i < \gamma} \tilde{\lambda}_i$
13. **if** $(|(s_i - s_{i-1})/s_{i-1}| <$ `tol`$)$ **then**
14. Calculate residuals $\|r_i\|$ and if all $\|r_i\| <$ `tol` **then break**
15. **end**
16. **end**

FIG. 4. *The filtered Lanczos algorithm. The inner product for vectors is denoted by* $\langle ., . \rangle$.

uniformly, on the (continuous) set $[\alpha, \beta]$ containing the spectrum (with $0 < \alpha < \beta$). A number of other methods have been developed which attempt to make the polynomial $\lambda s(\lambda)$ close to the function 1, in some least-squares sense, on the interval $[\alpha, \beta]$.

In the standard CR algorithm (see, e.g., [27]), the solution polynomial s_j minimizes the norm $\|(I - As(A))r_0\|_2$, which is nothing but a discrete least-squares norm when expressed in the eigenbasis of A:

$$\|(I - As(A))r_0\|_2 = \left[\sum_1^N (1 - \lambda_i s(\lambda_i))^2\right]^{1/2} \equiv \|1 - \lambda s(\lambda)\|_D.$$

It is possible to write a CR-like algorithm which minimizes $\|1 - \lambda s(\lambda)\|_g$ for any least-squares norm associated with a (proper) inner product of polynomials

$$\langle p, q \rangle_g.$$

The related generic CR algorithm is given in Figure 5.

It can be easily shown that the residual polynomial p_j generated by this algorithm minimizes $\|p(\lambda)\|_g$ among all polynomials of the form $p(\lambda) = 1 - \lambda s(\lambda)$, where s is any polynomial of degree $\leq j - 1$. In other words, p_j minimizes $\|p(\lambda)\|_g$ among all

GENERIC CONJUGATE RESIDUAL ALGORITHM.

1. Compute $r_0 := b - Ax_0$, $p_0 := r_0$, $\pi_0 = p_0 = 1$
2. Compute $\lambda\pi_0$
3. **for** $j = 0, 1, \ldots,$ until convergence:
4. $\alpha_j := \langle p_j, \lambda p_j\rangle_g / \langle \lambda\pi_j, \lambda\pi_j\rangle_g$
5. $x_{j+1} := x_j + \alpha_j p_j$
6. $r_{j+1} := r_j - \alpha_j A p_j$ $p_{j+1} = p_j - \alpha_j \lambda\pi_j$
7. $\beta_j := \langle p_{j+1}, \lambda p_{j+1}\rangle_g \langle p_j, \lambda p_j\rangle_g$
8. $p_{j+1} := r_{j+1} + \beta_j p_j$ $\pi_{j+1} := p_{j+1} + \beta_j \pi_j$
9. Compute $\lambda\pi_{j+1}$
10. **end**

FIG. 5. *Generic CR algorithm.*

polynomials p of degree $\leq j$, such that $p(0) = 1$. In addition, the polynomials $\lambda\pi_j$ are orthogonal to each other.

In order to add filtering to the above algorithm, note that filtering amounts to minimizing some norm of $\phi(\lambda) - \lambda s(\lambda)$, where ϕ is the given filter function. One must remember that $\phi(A)v$ is not necessarily easy to evaluate for a given vector v. In particular, $\phi(A)r_0$ may not be available.

The relation between regularized filtered iterations and polynomial iterations, such as the one we are seeking for the eigenvalue problem, may not be immediately clear. Observe that the residual polynomial $p_m(t)$ can be used as a filter polynomial for a given iteration. For example, the residual polynomial shown on the right of Figure 7, which is of the form $p(\lambda) = 1 - \lambda s(\lambda)$, can be used for computing all eigenvalues in the interval $[0, 1.7]$. The dual filter $1 - p(\lambda)$ has small values in $[0, 1.7]$ and can be used to compute the invariant subspace associated with the eigenvalues in the interval $[2.3, 8]$, though this may possibly require a large subspace. Notice that one of the main difficulties with this class of techniques is precisely the issue of the dimension of the subspace, as there is no inexpensive way of knowing in advance how many eigenvalues there are in a given interval.

4.1. Corrected CR algorithm. The standard way of computing the best polynomial is to generate an orthogonal sequence of polynomials and expand the least-squares solution in it. This approach was taken in [6] and more recently in [12].

The formulation of the solution given next is based on the following observation. The polynomials associated with the residual vectors of the (standard) CR algorithm are such that $\{\lambda\pi_j\}$ is an orthogonal sequence of polynomials, and so it can be used as an intermediate sequence in which to express the solution. We can generate the residual polynomial which will help obtain the p_i's: [illegible], i.e., the same r vectors as those of the generic CR algorithm (see Figure 5). It is interesting to note that with this sequence of residual vectors, which will be denoted by $\tilde{r}_j$, it is easy to generate the directions p_i [illegible] for both algorithms. The idea becomes straightforward: obtain the auxiliary residual polynomials $\tilde{p}_j$ that are those associated with the [illegible] CR algorithm and exploit them to obtain the π_i's in the same way as in the CR algorithm. The polynomials $\lambda\pi_j$ are orthogonal and therefore the expression of the desired approximation is the same. The algorithm is described in Figure 6, where now $\tilde{p}_j$ is the polynomial associated with the auxiliary sequence $\tilde{r}_j$.

FILTERED CONJUGATE RESIDUAL POLYNOMIALS ALGORITHM.

1.	Compute $\tilde{r}_0 := b - Ax_0$, $p_0 := \tilde{r}_0$	$\pi_0 = \tilde{p}_0 = 1$
2.		Compute $\lambda\pi_0$
3.	**for** $j = 0, 1, \ldots,$ until convergence:	
4.	$\tilde{\alpha}_j := \langle \tilde{p}_j, \lambda\tilde{p}_j \rangle_w / \langle \lambda\pi_j, \lambda\pi_j \rangle_w$	
5.	$\alpha_j := \langle \phi, \lambda\pi_j \rangle_w / \langle \lambda\pi_j, \lambda\pi_j \rangle_w$	
6.	$x_{j+1} := x_j + \alpha_j p_j$	
7.	$\tilde{r}_{j+1} := \tilde{r}_j - \tilde{\alpha}_j A p_j$	$\tilde{p}_{j+1} = \tilde{p}_j - \tilde{\alpha}_j \lambda\pi_j$
8.	$\beta_j := \langle \tilde{p}_{j+1}, \lambda\tilde{p}_{j+1} \rangle_w / \langle \tilde{p}_j, \lambda\tilde{p}_j \rangle_w$	
9.	$p_{j+1} := r_{j+1} + \beta_j p_j$	$\pi_{j+1} := \tilde{p}_{j+1} + \beta_j \pi_j$
10.		Compute $\lambda\pi_{j+1}$
11.	**end**	

FIG. 6. *The filtered CR polynomials algorithm.*

The only difference with a generic CR-type algorithm (see, e.g., Figure 5) is that the updates to x_{j+1} use different coefficients α_j from the updates to the vectors $\tilde{r}_{j+1}$. Observe that the residual vectors $\tilde{r}_j$ obtained by the algorithm are just auxiliary vectors that do not correspond to the actual residuals $r_j = b - Ax_j$. Needless to say, these actual residuals, the r_j's, can also be generated after line 5 (or 6) from $r_{j+1} = r_j - \alpha_j A p_j$. Depending on the application, it may or may not be necessary to include these computations.

The solution vector x_{j+1} computed at the jth step of the corrected CR algorithm is of the form $x_{j+1} = x_0 + s_j(A)r_0$, where s_j is the jth degree polynomial:

$$s_j(\lambda) = \alpha_0\pi_0(\lambda) + \cdots + \alpha_j\pi_j(\lambda). \tag{15}$$

The polynomials π_j and the auxiliary polynomials $\tilde{p}_{j+1}(\lambda)$ satisfy the orthogonality relations,

$$\langle \lambda\pi_j(\lambda), \lambda\pi_i(\lambda) \rangle_w = \langle \lambda\tilde{p}_j(\lambda), \tilde{p}_i(\lambda) \rangle_w = 0 \quad \text{for} \quad i \neq j. \tag{16}$$

In addition, the filtered residual polynomial $\phi - \lambda s_j(\lambda)$ minimizes $\|\phi - \lambda s(\lambda)\|_w$ among all polynomials s of degree $\leq j - 1$.

It is worth mentioning that there is an alternative formula for α_j, which is

$$\alpha_j = \tilde{\alpha}_j - \frac{\langle 1 - \phi, \lambda\pi_j \rangle}{\langle \lambda\pi_j, \lambda\pi_j \rangle}, \tag{17}$$

whose merit, relative to the expression used in line 4 of the algorithm, is that it clearly establishes the new algorithm as a corrected version of the generic CR algorithm of Figure 5. In the special situation when $\phi \equiv 1$, $\alpha_i = \tilde{\alpha}_i$, and the two algorithms coincide as expected.

4.2. The base filter function. The solutions computed by the algorithms just seen consist of generating polynomial approximations to a certain base filter function ϕ. It is generally not a good idea to use ϕ as the step function because this function is discontinuous and approximations to it by high-degree polynomials will exhibit very wide oscillations near the discontinuities. It is preferable to take as a "base" filter, i.e., the filter which is ultimately approximated by polynomials, a smooth function such as the one illustrated in Figure 7.

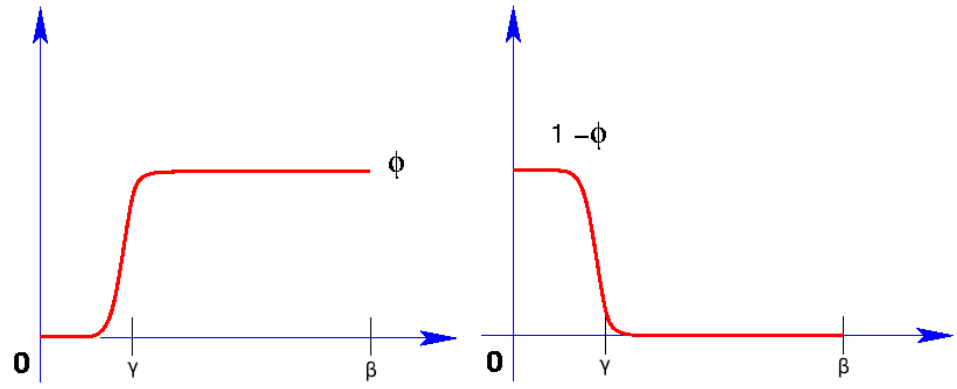


FIG. 7. *A typical filter function ϕ and its dual filter $1-\phi$.*

The filter function in Figure 7 can be a piecewise polynomial consisting of two pieces: A function which increases from 0 to 1 when λ increases smoothly from 0 to γ, and the constant function unity in the interval $[\gamma, \beta]$. Alternatively, the function can begin with the value 0 in the interval $[0, \gamma_1]$, then increase smoothly from 0 to 1 in a second interval $[\gamma_1, \gamma_2]$, and finally take the value 1 in $[\gamma_2, \beta]$. This second part of the function (the first part for the first scenario) bridges the values 1 and 1 by a smooth function and was termed a "bridge function" in [6].

A systematic way of generating base filter functions is to use bridge functions obtained from Hermite interpolation. The bridge function is an interpolating polynomial (in the Hermite sense) depending on two integer parameters m_0, m_1 and denoted by $\Theta_{[m_0,m_1]}$ which satisfies the following conditions:

$$
\begin{aligned}
&\Theta_{[m_0,m_1]}(0) = 0; \quad \Theta'_{[m_0,m_1]}(0) = \cdots = \quad \Theta^{(m_0)}_{[m_0,m_1]}(0) = 0, \\
&\Theta_{[m_0,m_1]}(\gamma) = 1; \quad \Theta'_{[m_0,m_1]}(\gamma) = \cdots = \quad \Theta^{(m_1)}_{[m_0,m_1]}(\gamma) = 0.
\end{aligned} \tag{18}
$$

Thus, $\Theta_{[m_0,m_1]}$ has degree m_0+m_1+1 and m_0, m_1 define the degree of smoothness at the points 0 and α, respectively. The ratio $\frac{m_1}{m_0}$ determines the localization of the inflection point. Making the polynomial increase rapidly from 0 to 1 in a small interval can be achieved by taking high-degree polynomials, but this has the effect of slowing down convergence toward the desired filter, as it causes undesired oscillations. Two examples are shown in Figures 8 and 9.

Once the base filter is selected, the filtered CR algorithm can be executed. It remains, however, to define the inner products. Details on the weight functions and the actual techniques for computing inner products of polynomials can be found in [28]. We only mention that it is possible to avoid numerical integration by defining the inner products by using classical weights (e.g., Chebyshev) in each subinterval of the whole interval where the base filter is defined. Since the base filter is a standard polynomial in each of these subintervals, inner products in these intervals can be evaluated without numerical integration. This, in effect, is equivalent to using Gaussian quadrature in each of these subintervals.

The support of the bridge function, an interval in which the base function drops from 1 to 0, can be determined by the interval of wanted eigenvalues and the largest eigenvalue of the matrix. We already discussed how to get the required bounds for the largest and smallest eigenvalues of A from a few steps of the Lanczos algorithm.

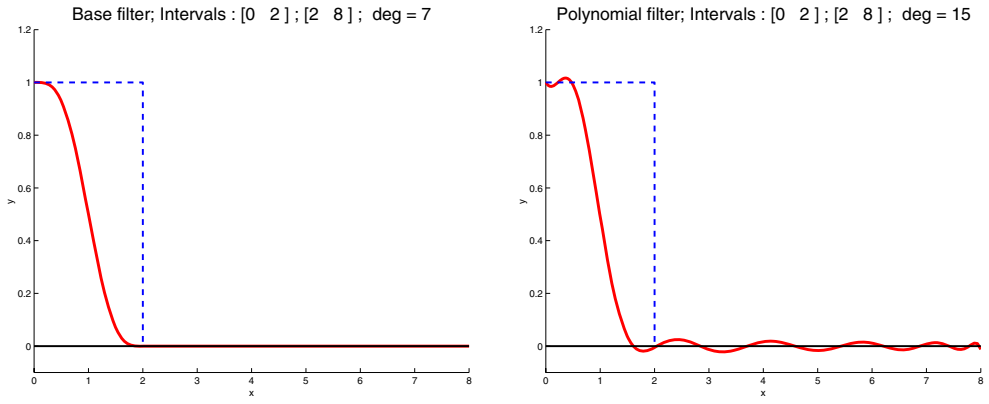


FIG. 8. *The base filter* $\Theta_{[4,4]}$ *in* $[0,2]$ *and one in* $[2,8]$ *(left) and its polynomial approximation of degree* 15 *(right).*

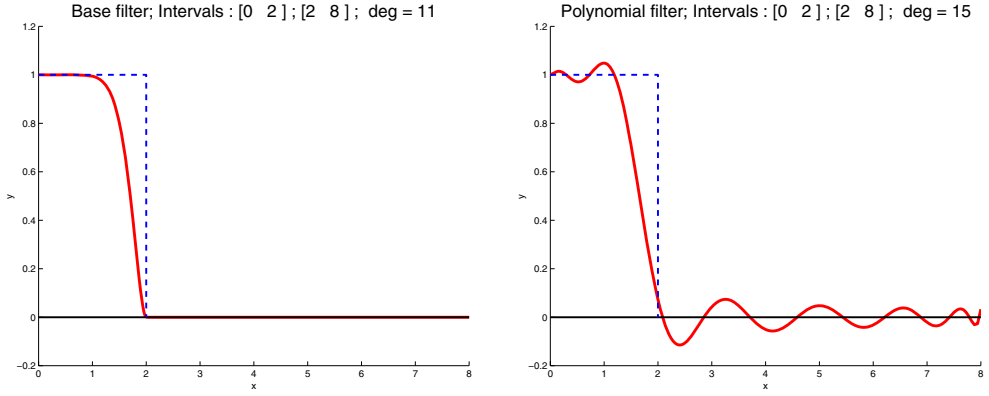


FIG. 9. *The base filter* $\Theta_{[10,2]}$ *in* $[0,2]$ *and one in* $[2,8]$ *(left) and its polynomial approximation of degree* 15 *(right).*

The bridge interval should be large enough so as to include all the eigenvalues of the wanted set, but not so large as to contain too many unwanted eigenvalues. In practice, augmenting the interval of wanted eigenvalues slightly only minimally hampers performance but helps improve the robustness of the procedure.

There are a number of parameters which can be exploited to yield a desired filter polynomial. In addition to the degrees of the polynomials m_0, m_1, one can also define the weight functions differently. For example, more or less emphasis can be placed in each subinterval. Our experience shows that using an equal weighting scheme for each subinterval is a very reasonable choice for most applications, including electronic structure calculations.

5. Numerical experiments. This section reports on a few numerical experiments with matrices taken from electronic structure calculations and from the Harwell–Boeing collection. Two other good reference points for a useful comparison would be the partially reorthogonalized Lanczos (which was used in [1]) and the implicitly restarted Lanczos iteration as it is implemented in the popular package `ARPACK` [16, 33]. We compare these two algorithms with the filtered Lanczos (`F. Lanczos`) algorithm with partial reorthogonalization.

All the experiments which follow were performed on an SGI Origin 2000 system using a single R12000 processor at 300 MHz clock. The filtered Lanczos code is available from the authors upon request. `F. Lanczos` is implemented purely in C while `ARPACK` is implemented in Fortran 77. The Lanczos algorithm with partial reorthogonalization is based in the Fortran 77 code `PLANSO` [35]. The convergence tolerance was set to 10^{-10} for all methods. Notice that although in electronic structure calculations the convergence tolerance is typically taken 2–3 orders of magnitude larger, `F. Lanczos` (similar to `ARPACK` and Lanczos) can be used for other applications as well (this is why we include the test matrix from the Harwell–Boeing collection). In order to conduct a rather strict test we have chosen the above convergence tolerance. For `ARPACK` the maximum dimension of the Lanczos basis was always set equal to twice the number of requested eigenvalues. Thus, the number of implicit QR steps in `ARPACK` was equal to the number of the wanted eigenvalues. We point out that these settings are typically used in `ARPACK`.

For `F. Lanczos` the number of filtered Lanczos iterations for the initial vector was set to 200, while the degree of the inner CR polynomial was 8. In the latest (stabilized) version of the code we use 2 intervals for the base function: one for the wanted and another for the unwanted ones. The degrees m_1, m_2 for the smooth base function are set to $m_1 = 5$ and $m_2 = 15$. The number of Lanczos iterations for the determination of the bounding interval $[\alpha, \beta]$ for the spectrum was determined by a convergence tolerance of 10^{-6}. The above settings were the same for all test cases.

For partial reorthogonalization we used the default parameters defined in the `PLANSO` code. It is worth mentioning that the maximum loss of orthogonality allowed was set to the square root of the machine precision.

In implicitly restarted techniques, such as those implemented in `ARPACK`, a basis of length equal to the number of required eigenvalues is updated at each restart. Thus, such methods are not designed to compute all eigenvalues in a given interval. This, of course, is in contrast to the filtered Lanczos iteration, as well as to the unrestarted Lanczos algorithm. In order to facilitate a performance comparison we have used the following setting: for each test matrix, we are interested in a given number of its algebraically smallest eigenvalues. We compute these using `ARPACK`. Then we use the filtered Lanczos iteration and the unrestarted Lanczos iteration with partial reorthogonalization to compute all eigenvalues that are smaller than or equal to the largest of the requested eigenvalues computed by `ARPACK`. Of course, this comparison is not carried out on completely equal terms. However, our goal is to demonstrate that a strategy of exchanging memory accesses with additional matrix-vector products can significantly lower the overall computational cost. This was previously shown in [1], however, at the important expense of additional memory, relative to implicitly restarted techniques. The experiments that follow clearly show that the filtered Lanczos iteration can achieve both goals: it can operate on limited memory while significantly reducing the overall computational cost.

Test matrices. We have used four matrices from electronic structure calculations for the tests. These are Hamiltonians obtained from a real-space code [2]. In addition, we have also used a test matrix, namely, the `Andrews` matrix, from the University of Florida sparse matrix collection,[3] so as to give an example of applicability of our method in other applications as well. Table 1 provides the characteristics of the test matrices. For the Hamiltonians the number of the requested eigenvalues

[3] http://www.cise.ufl.edu/research/sparse/

TABLE 1
Characteristics of test matrices: nnz is the total number of nonzeros, so the last column shows the average number of nonzeros per row.

Matrix	Size n	nnz	nnz/n
$\mathtt{Si_{10}H_{16}}$	17077	875923	51.3
$\mathtt{Ge_{99}H_{100}}$	94341	6332795	67.2
$\mathtt{Ge_{87}H_{76}}$	94341	5963003	63.2
$\mathtt{Si_{34}H_{36}}$	97569	5156379	52.8
`Andrews`	60000	760154	12.7

generally correspond to physical properties of the corresponding molecular system. For example, $\mathtt{Si_{10}H_{16}}$ has 28 occupied states, while $\mathtt{Si_{34}H_{36}}$ has 86, $\mathtt{Ge_{87}H_{76}}$ has 212, and $\mathtt{Ge_{99}H_{100}}$ has 248. In order to test the scalability of the methods under study, we requested additional eigenvalues as well. For the matrix `Andrews` we arbitrarily requested 100–400 eigenvalues. We point out that all statistics for the `F. Lanczos` algorithm include an initial call to the unrestarted Lanczos algorithm, with partial reorthogonalization, in order to approximate (upper and lower) bounds for the extremal eigenvalues. Observe that our choice of matrices spans different degrees of sparsity in order to demonstrate the effect of the latter on the overall cost, since the `F. Lanczos` algorithm makes heavy use of matrix-vector products.

Discussion. The experimental results clearly illustrate that the `F. Lanczos` algorithm achieves significant improvements over the other two competing methods. The performance improvement becomes more evident as the number of requested eigenvalues increases.

All of our test matrices are sparse. However, the degree of sparsity (as measured by the average number of nonzeros per row, shown in the last column of Table 1) differs significantly between the "denser" $\mathtt{Ge_{99}H_{100}}$ Hamiltonian and the "sparser" `Andrews` matrix. A careful look in the results illustrated in Table 2 clearly suggests that the improvements in run-times of the `F. Lanczos` algorithm over `ARPACK` is more pronounced for the sparser test matrices. Thus, although the number of matrix-vector products in `F. Lanczos` increases relative to `ARPACK`, a significant gain results from avoiding the updating of a large number of eigenvectors, which standard methods do at every step.

The use of partial reorthogonalization is indeed beneficial in both `F. Lanczos` and Partial Lanczos. However, the main advantage of `F. Lanczos` is the reduction of traffic in memory. For example, let us look at the case of $\mathtt{Si_{34}H_{36}}$ and $n_o = 200$ (last row of subtable, Table 2). Observe that `F. Lanczos` uses 640 basis vectors, while Partial Lanczos has to move 3580 basis vectors from memory. For `ARPACK` we have 30 restarts, and at each restart the algorithm will "touch" 400 vectors (twice the number of sought eigenvalues); thus we have a total of at least 12000 basis vectors moving between memory and CPU (including other costs such as full reorthogonalization).

In electronic structure calculations the required accuracy is close to $0.5 \cdot 10^{-6}$, which is larger than the semiorthogonality level of 10^{-8} that is ensured by partial reorthogonalization. However, in applications that have stricter accuracy requirements, semiorthogonality of basis vectors may not be adequate. If this is the case, then we can use full reorthogonalization in `F. Lanczos`. Of course, we can expect the benefits over `ARPACK` to reduce somewhat; however, the major improvement which results from the small basis of `F. Lanczos` and its unrestarted nature, and thus its much lesser use of memory, is still there. On the other hand, using a large convergence tolerance could prove tricky in `ARPACK` as this code relies on deflation techniques in order to improve

TABLE 2

Summary of experimental results for all 5 *test matrices.* MV *denotes the total number of matrix-vector products, which for the Lanczos algorithm with partial reorthogonalization is also the dimension of the Lanczos basis used. For the* F. Lanczos *algorithm, the numbers in parentheses in the* MV *column denote the dimension of the Lanczos basis.* RTH *denotes the number of reorthogonalization steps.* RES *is the number of restarts for* ARPACK. MEM *denotes the required memory in Mbytes and* t *is the total time in secs. Finally, n_o is the number of requested eigenvalues.*

Andrews												
	F. Lanczos				Partial Lanczos				ARPACK			
n_o	MV	RTH	MEM	t	MV	RTH	MEM	t	MV	RES	MEM	t
100	3320 (290)	130	133	330	1390	111	636	530	1616	24	92	2000
200	6110 (600)	186	275	803	2360	213	1080	1633	2769	21	183	6682
300	8270 (840)	224	385	1364	3120	298	1428	2976	3775	19	275	13572
400	10610 (1100)	267	504	2274	3970	393	1817	4997	4978	19	366	23762

$Si_{10}H_{16}$												
	F. Lanczos				Partial Lanczos				ARPACK			
n_o	MV	RTH	MEM	t	MV	RTH	MEM	t	MV	RES	MEM	t
28	1144 (100)	21	13	48	539	16	72	24	592	27	7.5	58
50	1864 (180)	40	23	86	930	35	124	61	1039	31	13.3	187
150	4384 (460)	86	60	244	1940	97	259	273	2129	21	40	1111
200	5284 (560)	88	73	315	2190	114	292	360	2676	20	53	1847

$Si_{34}H_{36}$												
	F. Lanczos				Partial Lanczos				ARPACK			
n_o	MV	RTH	MEM	t	MV	RTH	MEM	t	MV	RES	MEM	t
86	2317 (230)	42	171	778	1440	36	1098	605	1537	24	131	2877
100	3127 (320)	54	238	1105	1810	50	1380	907	2164	32	152	4800
150	4657 (490)	102	365	1799	2880	96	2195	2191	3085	32	229	9993
200	6007 (640)	134	476	2496	3580	129	2729	3431	3803	30	305	16099

$Ge_{87}H_{76}$												
	F. Lanczos				Partial Lanczos				ARPACK			
n_o	MV	RTH	MEM	t	MV	RTH	MEM	t	MV	RES	MEM	t
212	4476 (470)	88	338	1895	2710	88	1951	1993	2867	20	306	12145
300	8256 (890)	172	641	4130	4010	153	2887	4448	4673	25	432	28359
424	11406 (1240)	240	893	6624	5740	252	4132	9804	6059	23	611	51118

$Ge_{99}H_{100}$												
	F. Lanczos				Partial Lanczos				ARPACK			
n_o	MV	RTH	MEM	t	MV	RTH	MEM	t	MV	RES	MEM	t
248	5194 (550)	102	396	2379	3150	109	2268	2746	3342	20	357	16454
350	8794 (950)	178	684	4648	4570	184	3289	5982	5283	24	504	37371
496	12934 (1410)	270	1015	8374	6550	302	4715	13714	6836	22	714	67020

convergence. Thus, poorly converged (large) eigenvectors will reenter in the iteration, slowing convergence towards small eigenvalues.

In comparison with the unrestarted partially reorthogonalized Lanczos procedure, observe that the filtered Lanczos method always requires far less memory. In fact, the amount of additional memory in comparison to ARPACK is quite modest. Typically, the new method will require a Lanczos basis with length close to three times the number of computed eigenvalues. We also observe that for rather dense matrices and small number of eigenvalues (i.e., $Si_{34}H_{36}$ and $Si_{10}H_{16}$) the unrestarted Lanczos method with partial reorthogonalization is the fastest of the three methods. However, when a large invariant subspace is sought, then the unrestarted Lanczos method will tend to require a long basis, ultimately causing even infrequent reorthogonalizations and a significant increase in memory traffic, and to dramatically prolong the run-times.

6. Conclusions. This paper presented a filtered Lanczos iteration for computing large invariant subspaces associated with the algebraically smallest eigenvalues of very large and sparse matrices. In contrast to restarted techniques (e.g., ARPACK),

which repeatedly update a fixed number of basis vectors, filtered Lanczos is allowed to augment the search subspace until all eigenvalues smaller than a predetermined upper bound have converged. The loss of orthogonality of the Lanczos basis vectors is treated by a partial reorthogonalization scheme [31]. One technique which filtered Lanczos and explicit/implicit restarted Krylov subspace algorithms have in common is the use of filtering polynomials, designed to dampen eigencomponents along "unwanted" parts of the spectrum. However, while restarted techniques apply these polynomials periodically (i.e., at each restart), the filtered Lanczos procedure applies a fixed, precomputed, low-degree polynomial of A to the working Lanczos vector, which amounts to a polynomial preconditioning technique applied to A. We showed that if the unwanted eigendirections are thoroughly filtered from the starting vector of the Lanczos algorithm, then the application of the aforementioned small-degree polynomial successfully prevents the unwanted directions from reappearing into the iteration, thus expediting convergence towards the desired invariant subspace. Earlier work (see, e.g., [28]) showed how one can design a CR-type iteration that efficiently applies a low pass filter in order to solve regularized linear systems. The low-degree polynomial which is involved in this procedure is used in the filtered Lanczos algorithm.

Experimental evidence clearly shows that the new method achieves significant performance improvements over the most sophisticated restarted technique (i.e., `ARPACK`), while at the same time incurring very modest additional memory requirements. These gains in efficiency are obtained by essentially trading the repeated and costly updates of the working eigenbasis, which is inherit in restarted techniques, for additional matrix-vector products. Thus, the method will work quite well whenever matrix-vector products are not expensive.

Acknowledgments. This work would not have been possible without the availability of excellent source codes for diagonalization. Specifically, our experiments made use of the `PLANSO` code developed by Wu and Simon [35] and the `ARPACK` code of Lehoucq, Sorensen, and Yang [16]. The first author would like to thank A. Stathopoulos, R. Lehoucq, and C. Yang for useful discussions concerning implicitly restarted methods.

REFERENCES

[1] C. Bekas, Y. Saad, M. Tiago, and J. Chelikowsky, *Computing charge densities with partially reorthogonalized Lanczos*, Comput. Phys. Comm., 171 (2005), pp. 175–186.

[2] J. R. Chelikowsky, L. Kronik, I. Vasiliev, M. Jain, and Y. Saad, *Using real space pseudopotentials for electronic structure calculations*, in Handbook of Numerical Analysis, Vol. 10, Special Volume on Computational Chemistry, C. Le Bris, ed., Elsevier Science B.V., Amsterdam, 2003, pp. 613–637.

[3] J. Cullum and R. A. Willoughby, *Lánczos Algorithms for Large Symmetric Eigenvalue Computations*, Vol. 1, Birkhäuser, Boston, 1985.

[4] J. Cullum and R. A. Willoughby, *Lánczos Algorithms for Large Symmetric Eigenvalue Computations*, Vol. 2, Birkhäuser, Boston, 1985.

[5] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.

[6] J. Erhel, F. Guyomarc, and Y. Saad, *Least-Squares Polynomial Filters for Ill-Conditioned Linear Systems*, Technical Report umsi-2001-32, Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, 2001.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[8] P. Hohenberg and W. Kohn, *Inhomogeneous electron gas*, Phys. Rev., 136 (1965), pp. B864–B871.

[9] Y. Huang, D. J. Kouri, and D. K. Hoffman, *Direct approach to density functional the-*

ory: Iterative treatment using a polynomial representation of the heaviside step function operator, Chem. Phys. Lett., 243 (1995), pp. 367–377.

[10] L. O. Jay, H. Kim, Y. Saad, and J. R. Chelikowsky, *Electronic structure calculations using plane wave codes without diagonalization*, Comput. Phys. Comm., 118 (1999), pp. 21–30.

[11] W. Kohn and L. J. Sham, *Self-consistent equations including exchange and correlation effects*, Phys. Rev., 140 (1965), pp. A1133–A1138.

[12] E. Kokiopoulou and Y. Saad, *Polynomial filtering in latent semantic indexing for information retrieval*, in Proceedings of the ACM-SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 2004, pp. 104–111.

[13] G. Kresse and J. J. Furthmüller, *Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set*, Phys. Rev. B, 54 (1996), pp. 11169–11186.

[14] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Research Nat. Bur. Standards, 45 (1950), pp. 255–282.

[15] R. M. Larsen, *Efficient Algorithms for Helioseismic Inversion*, Ph.D. thesis, Department of Computer Science, University of Aarhus, Aarhus, Denmark, 1998.

[16] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998, http://www.caam.rice.edu/software/ARPACK.

[17] W. Z. Liang, C. Saravanan, Y. Shao, A. Bell, and H. Head-Gordon, *Improved Fermi operator expansion methods for fast electronic structure calculations*, J. Chem. Phys., 119 (2003), pp. 4117–4125.

[18] F. Mauri and G. Galli, *Electronic-structure calculations and molecular-dynamics simulation with linear system size scaling*, Phys. Rev. B, 50 (1994), pp. 4316–4326.

[19] F. Mauri, G. Galli, and R. Car, *Orbital formulation for electronic-structure calculations with linear system size scaling*, Phys. Rev. B, 47 (1993), pp. 9973–9976.

[20] A. Neumaier, *Solving ill-conditioned and singular linear systems: A tutorial on regularization*, SIAM Rev., 40 (1998), pp. 636–666.

[21] A. M. N. Niklasson and M. Challacombe, *Density matrix perturbation theory*, Phys. Rev. Lett., 92 (2004), article 193001.

[22] A. M. N. Niklasson, C. J. Tymczak, and M. Challacombe, *Trace resetting density matrix purification in O(n) self-consistent-field theory*, J. Chem. Phys., 118 (2003), pp. 8611–8620.

[23] C. C. Paige, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph.D. thesis, London University, London, 1971.

[24] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998.

[25] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos, *Iterative minimization techniques for ab-initio total energy calculations: Molecular dynamics and conjugate gradients*, Rev. Modern Phys., 64 (1992), pp. 1045–1097.

[26] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Halstead Press, New York, 1992.

[27] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[28] Y. Saad, *Filtered conjugate residual-type algorithms with applications*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 845–870.

[29] Y. Saad, A. Stathopoulos, J. Chelikowsky, K. Wu, and S. Öğüt, *Solution of large eigenvalue problems in electronic structure calculations*, BIT, 36 (1996), pp. 563–578.

[30] A. H. Sameh and J. A. Wisniewski, *A trace minimization algorithm for the generalized eigenvalue problem*, SIAM J. Numer. Anal., 19 (1982), pp. 1243–1259.

[31] H. D. Simon, *Analysis of the symmetric Lanczos algorithm with reorthogonalization methods*, Linear Algebra Appl., 61 (1984), pp. 101–132.

[32] H. D. Simon, *The Lanczos algorithm with partial reorthogonalization*, Math. Comp., 42 (1984), pp. 115–142.

[33] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[34] G. W. Stewart, *Adjusting the Rayleigh quotient in semiorthogonal Lanczos methods*, SIAM J. Sci. Comput., 24 (2002), pp. 201–207.

[35] K. Wu and H. Simon, *A Parallel Lanczos Method for Symmetric Generalized Eigenvalue Problems*, Report 41284, Lawrence Berkeley National Laboratory, 1997.

[36] C. Yang, J. Meza, and L. W. Wang, *A constrained optimization algorithm for total energy minimization in electronic structure calculation*, J. Comput. Phys., 217 (2006), pp. 709–721.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 419–443

A FAST METHOD FOR FINDING THE GLOBAL SOLUTION OF THE REGULARIZED STRUCTURED TOTAL LEAST SQUARES PROBLEM FOR IMAGE DEBLURRING*

AMIR BECK†, AHARON BEN-TAL‡, AND CHRISTIAN KANZOW§

Abstract. Given a linear system $\mathbf{Ax} \approx \mathbf{b}$ over the real or complex field, where both $\mathbf{A}$ and $\mathbf{b}$ are subject to noise, the total least squares (TLS) problem seeks to find a correction matrix and a correction right-hand side vector of minimal norm which makes the linear system feasible. To avoid ill posedness, a regularization term is added to the objective function; this leads to the so-called regularized TLS problem. A further complication arises when the matrix $\mathbf{A}$ and correspondingly the correction matrix must have a specific structure. This is modeled by the regularized structured TLS (RSTLS) problem. In general this problem is nonconvex and hence difficult to solve. However, the RSTLS problem arising from image deblurring applications under reflexive or periodic boundary conditions possesses a special structure where all relevant matrices are simultaneously diagonalizable (SD). In this paper we introduce an algorithm for finding the *global* optimum of the RSTLS problem with this SD structure. The devised method is based on decomposing the problem into single variable problems and then transforming them into one-dimensional unimodal real-valued minimization problems which can be solved globally. Based on the uniqueness and attainment properties of the RSTLS solution we show that a constrained version of the problem possesses a strong duality result and can thus be solved via a sequence of RSTLS problems.

Key words. structured total least squares, nonconvex optimization, image deblurring, unimodal functions, simultaneously diagonalizable matrices

AMS subject classifications. 90C26, 15A29

DOI. 10.1137/070709013

1. Introduction.

Given a linear system $\mathbf{Ax} \approx \mathbf{b}$ over the real or complex field, where both the matrix $\mathbf{A}$ and the right-hand side vector $\mathbf{b}$ are subjected to noise, the total least squares (TLS) problem seeks to minimize the sum of squared norms of the perturbations to both the model matrix and vector $\|\mathbf{E}\|^2 + \|\mathbf{w}\|^2$ subject to the condition that the perturbed system holds: $(\mathbf{A} + \mathbf{E})\mathbf{x} = \mathbf{b} + \mathbf{w}$. Although this problem is nonconvex, it can be solved efficiently and globally by using a spectral decomposition of the augmented matrix $(\mathbf{A}, \mathbf{b})$; see [14, 20].

In many applications, the matrix $\mathbf{A}$ has a specific linear structure, e.g., Toeplitz or Hankel, which imposes a requirement on the perturbation matrix $\mathbf{E}$ to possess a corresponding special structure. The TLS solution does not take into account this requirement, and consequently the structured TLS (STLS)[1] attracted intensive research; see, e.g., [1, 29, 34, 28, 25, 22]. The formulation of the STLS problem is

*Received by the editors November 26, 2007; accepted for publication (in revised form) by N. Mastronardi January 22, 2008; published electronically April 23, 2008.
http://www.siam.org/journals/simax/30-1/70901.html

†Department of Industrial Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (becka@ie.technion.ac.il). This author's research was partly supported by GIF Young Scientists' Research grant 1542/2005.

‡MINERVA Optimization Center, Department of Industrial Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (abental@ie.technion.ac.il). This author's research was partly supported by the Technion VPR fund for promotion of research, grant 2005519.

§Institute of Mathematics, University of Würzburg, Am Hubland, 97074 Würzburg, Germany (kanzow@mathematik.uni-wuerzburg.de).

[1]In some papers the STLS problem is also called *constrained total least squares*.

$$\text{(STLS):}\quad \begin{array}{ll} \min\limits_{\mathbf{E},\mathbf{x},\mathbf{w}} & \|\mathbf{E}\|^2 + \|\mathbf{w}\|^2 \\ \text{s.t.} & (\mathbf{A}+\mathbf{E})\mathbf{x} = \mathbf{b}+\mathbf{w}, \\ & \mathbf{E} \in \mathcal{L}, \end{array}$$

where $\mathcal{L}$ is a linear subspace. We remark that there are several generalizations of the above STLS formulation that are able to deal with multiple right-hand sides (that is, $\mathbf{b}$ and $\mathbf{x}$ are matrices) [23], structure of the right-hand side noise vector $\mathbf{w}$ [23], and other norms such as l_1, l_∞ [34], and weighted l_2 norms [24].

The STLS problem is a nonconvex problem, and thus finding its global solution is in general a difficult task. There are only a few exceptions to this state of affairs. For block circulant structures with unstructured blocks the corresponding STLS problem can be solved by decomposing the problem into several smaller TLS problems using the discrete Fourier transform [6]. Another tractable case arises when some of the columns of $\mathbf{A}$ are error-free while the others are subjected to noise. This problem is called the [illegible] problem or [illegible] problem, and its solution can be obtained by computing a QR factorization of $\mathbf{A}$ and then solving a TLS problem of reduced dimension [19]. A more general problem is the [illegible] problem introduced in [21]. There it is assumed that $(\mathbf{E}, \mathbf{w}) = \mathbf{D}_1 \tilde{\mathbf{E}} \mathbf{C}_1$, where $\mathbf{D}_1$ and $\mathbf{C}_1$ are known matrices and $\widetilde{\mathbf{E}}$ is unknown. As was shown in [21], by choosing the matrices $\mathbf{D}_1$ and $\mathbf{C}_1$ appropriately, the restricted TLS problem contains as special cases any weighted least squares (LS), generalized LS, TLS, and generalized TLS problems. The restricted TLS problem can be solved by using the restricted singular value decomposition [37].

In this paper we consider yet another tractable class of STLS problems in which the global solution can efficiently be found. We deal with structures in which all of the matrices in $\mathcal{L}$ are square and can be diagonalized by a certain fixed orthogonal (or unitary in the complex case) matrix. These structures are called [illegible] (SD) structures. The motivation for considering such structures stems from image deblurring problems with spatially invariant point spread functions (PSF). For two-dimensional image deblurring problems it is well known that the matrix describing the blur operator can be diagonalized by a two-dimensional discrete Fourier transform matrix when [illegible] are assumed. For [illegible] with symmetric PSF the corresponding matrix can be diagonalized by a two-dimensional discrete cosine transform matrix. Similar structures can be found in one-dimensional deconvolution problems. Section 2 contains a brief review of these structures.

A characteristic feature of image deblurring problems is that the matrix $\mathbf{A}$ is ill-conditioned, and as a result the STLS solution usually has a huge norm and as such is meaningless. Regularization is required in order to stabilize the solution. For the unstructured TLS problem several regularization methods are well known. Among them are truncation methods [11, 17] and Tikhonov regularization [13, 7], in which a quadratic penalty is added to the objective function or a quadratic constraint bounding the size of the solution norm is added to the problem [36, 33, 13, 8, 5].

For the STLS problem, Tikhonov regularization seems to be the most popular method. The resulting problem is called the regularized STLS problem (RSTLS) and is given by

$$\text{(RSTLS):}\quad \begin{array}{ll} \min\limits_{\mathbf{E},\mathbf{x},\mathbf{w}} & \|\mathbf{E}\|^2 + \|\mathbf{w}\|^2 + \rho\|\mathbf{L}\mathbf{x}\|^2 \\ \text{s.t.} & (\mathbf{A}+\mathbf{E})\mathbf{x} = \mathbf{b}+\mathbf{w}, \\ & \mathbf{E} \in \mathcal{L}. \end{array}$$

Common choices for $\mathbf{L}$ are the identity or a matrix approximating the first or second order derivative operator [16, 13, 18].

The RSTLS problem for structures arising in image deblurring was studied in several works. In [27] periodic boundary conditions are considered. By using the discrete Fourier transform the problem is decomposed into many complex-valued single-variable problems. The complex univariate problems are solved as two-variable nonconvex problems over the real domain by using the Davidon–Fletcher–Powell optimization algorithm.

In [31] an iterative algorithm of quasi-Newton form is applied for the RSTLS problem for reflexive boundary conditions that exploits the diagonalization properties of the associated matrices. The work [32] extends the structured total least norm algorithm [34] to include regularization, and image deblurring examples are discussed. This approach was also advocated in [12] for image deblurring problems with separable PSFs and in [26] for problems with zero boundary conditions.

In all of the above-mentioned works the optimization problems that need to be solved are nonconvex, and consequently the devised algorithms are not guaranteed to converge to a global optimum but rather to a stationary point. The main contribution of the present paper is the introduction of a method capable of obtaining the [illegible] of the RSTLS problem for SD structures.

The paper is organized as follows. In section 2 we present a precise problem formulation followed by a brief review of the essential ingredients from image deblurring. The decomposition of the RSTLS problem into single-variable real- or complex-valued problems is discussed in section 3. These univariate problems are not necessarily unimodal, but we show in section 4 that they can be transformed into single-variable real-valued unimodal problems. Attainment and uniqueness conditions are also obtained. In section 5 we concentrate on circulant structures and show that, when the data are real-valued, there exists at least one real-valued optimal solution (although the corresponding single-variable problems are complex-valued). In section 6 we tackle the constrained version of the RSTLS problem, called CSTLS, and show that, based on the derived uniqueness properties and on a strong duality result, the constrained problem can be solved by a sequence of RSTLS problems. The paper ends in section 7 with detailed descriptions of the numerical algorithms and a demonstration of our method as applied to an image deblurring problem. A MATLAB implementation and documentation of the RSTLS and CSTLS methods for image deblurring problems with either periodic or reflexive boundary conditions can be found in [38].

1.1. Notation. A vector or matrix is called real-valued (complex-valued) if all of its entries are real (complex). For a complex scalar a, the complex conjugate is denoted by $\bar{a}$. Given a matrix $\mathbf{A}$ (a vector $\mathbf{v}$), the complex conjugate is denoted by $\mathbf{A}^*$ ($\mathbf{v}^*$). For a real-valued matrix $\mathbf{Q}$, the complex conjugate $\mathbf{Q}^*$ translates to the usual transpose $\mathbf{Q}^T$, and unitarity translates to orthogonality: $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. The root of -1 is denoted by $\mathbf{i} = \sqrt{-1}$. For a given vector $\mathbf{v}$, $\|\mathbf{v}\|$ denotes the Euclidean norm of $\mathbf{v}$, and, for a matrix $\mathbf{A}$, $\|\mathbf{A}\|$ denotes the Frobenius norm of the matrix. The Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$.

2. RSTLS for simultaneously diagonalizable structures.

2.1. Problem formulation. The RSTLS problem can be written as follows:

$$\text{(RSTLS):} \quad \begin{array}{ll} \min & \|\mathbf{E}\|^2 + \|\mathbf{w}\|^2 + \rho\|\mathbf{L}\mathbf{x}\|^2 \\ \text{s.t.} & (\mathbf{A}+\mathbf{E})\mathbf{x} = \mathbf{b} + \mathbf{w}, \\ & \mathbf{E} \in \mathcal{L}, \\ & \mathbf{x} \in \mathbb{F}^n, \mathbf{w} \in \mathbb{F}^m, \end{array} \tag{2.1}$$

where $\mathbf{A} \in \mathbb{F}^{m\times n}$ and $\mathbf{b} \in \mathbb{F}^m$, with $\mathbb{F}$ being either the real or the complex number field ($\mathbb{R}$ or $\mathbb{C}$, respectively). The parameter ρ is a positive real number, and the set $\mathcal{L}$ is a linear subspace of the set of all $m \times n$ matrices $\mathbb{F}^{m\times n}$. As was discussed in the introduction, this formulation was considered in several papers; see, e.g., [27, 31, 32, 12, 26].

In this paper we consider the case in which $m = n$ and $\mathcal{L}$ is a linear subspace of the set of all $n \times n$ matrices diagonalizable by a given unitary matrix. That is, $\mathcal{L} = \mathcal{L}_{\mathbf{Q}}$, where

$$\mathcal{L}_{\mathbf{Q}} = \{\mathbf{Q}^* \mathrm{diag}(\boldsymbol{\lambda})\mathbf{Q} : \boldsymbol{\lambda} \in \mathbb{F}^n\}, \tag{2.2}$$

with $\mathbf{Q}$ being a given unitary matrix (i.e., $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$). Such a structure is called a SD structure, with unitary transform. In our derivations we also assume that $\mathbf{A}, \mathbf{L} \in \mathcal{L}_{\mathbf{Q}}$. This particular structure is also discussed in, e.g., [27, 31].

In section 3 we will show that, as opposed to most structures, the RSTLS problem with an SD structure can be solved globally and efficiently. Before doing so, we will describe some image deblurring examples in which SD structures appear.

2.2. SD structures associated with image deblurring. We will now present four classes of SD structures that arise naturally in image deblurring problems. In addition to two-dimensional images, we will also consider one-dimensional signals and refer to them as "one-dimensional images." Before examining the four classes, we briefly review some essential facts and notation from image processing.

Many image deblurring problems can be modeled as $\mathbf{g} = \mathbf{Sf}$, where $\mathbf{g} \in \mathbb{R}^n$ is the blurred image and $\mathbf{f} \in \mathbb{R}^n$ is the unknown true image, whose size is assumed to be the same as the one of $\mathbf{g}$. The matrix $\mathbf{S}$ describes the blur operator. In the case of spatially invariant blurs, $\mathbf{Sf}$ is usually a convolution of a corresponding PSF and the true image $\mathbf{f}$.

The structure of the matrix $\mathbf{S}$ depends on the choice of boundary conditions, that is, the underlying assumptions on the image outside the field of view. Three very popular boundary conditions are (i) [illegible], in which all pixels outside the borders are assumed to be zero, (ii) [illegible], in which it is assumed that the image repeats itself in all directions, (iii) [illegible], in which it is assumed that the scene outside of the boundaries is an image mirror of the image boundaries.

Let us illustrate the three types of boundary conditions. First, in the one-dimensional case consider the image

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix},$$

and then for zero, periodic, and reflexive boundary conditions the larger image looks like

$$\left(\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 1 \\ 2 \\ 3 \\ \hline 0 \\ 0 \\ 0 \end{array}\right), \qquad \left(\begin{array}{c} 1 \\ 2 \\ 3 \\ \hline 1 \\ 2 \\ 3 \\ \hline 1 \\ 2 \\ 3 \end{array}\right), \qquad \left(\begin{array}{c} 3 \\ 2 \\ 1 \\ \hline 1 \\ 2 \\ 3 \\ \hline 3 \\ 2 \\ 1 \end{array}\right),$$

respectively. In the two-dimensional case if we consider the image

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix},$$

then for zero, periodic, and reflexive boundary conditions the larger image looks like

$$\left(\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 5 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 8 & 9 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right), \quad \left(\begin{array}{ccc|ccc|ccc} 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \\ 4 & 5 & 6 & 4 & 5 & 6 & 4 & 5 & 6 \\ 7 & 8 & 9 & 7 & 8 & 9 & 7 & 8 & 9 \\ \hline 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \\ 4 & 5 & 6 & 4 & 5 & 6 & 4 & 5 & 6 \\ 7 & 8 & 9 & 7 & 8 & 9 & 7 & 8 & 9 \\ \hline 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \\ 4 & 5 & 6 & 4 & 5 & 6 & 4 & 5 & 6 \\ 7 & 8 & 9 & 7 & 8 & 9 & 7 & 8 & 9 \end{array}\right),$$

$$\left(\begin{array}{ccc|ccc|ccc} 9 & 8 & 7 & 7 & 8 & 9 & 9 & 8 & 7 \\ 6 & 5 & 4 & 4 & 5 & 6 & 6 & 5 & 4 \\ 3 & 2 & 1 & 1 & 2 & 3 & 3 & 2 & 1 \\ \hline 3 & 2 & 1 & 1 & 2 & 3 & 3 & 2 & 1 \\ 6 & 5 & 4 & 4 & 5 & 6 & 6 & 5 & 4 \\ 9 & 8 & 7 & 7 & 8 & 9 & 9 & 8 & 7 \\ \hline 9 & 8 & 7 & 7 & 8 & 9 & 9 & 8 & 7 \\ 6 & 5 & 4 & 4 & 5 & 6 & 6 & 5 & 4 \\ 3 & 2 & 1 & 1 & 2 & 3 & 3 & 2 & 1 \end{array}\right),$$

respectively. The structure of the matrix $\mathbf{S}$ depends on the underlying boundary conditions. Here we consider spatially invariant blurs which, as was already mentioned, imply that the blur is a convolution of given a PSF with the true (larger) image. For one-dimensional problems the PSF is just a vector $\mathbf{p} \in \mathbb{R}^d$ with an associated center $c \in \{1, 2, \ldots, d\}$. The convolution operation is then:

$$g_i = \sum_{j=1}^{d} p_j f_{i+c-j}, \quad i = 1, \ldots, n,$$

where $\mathbf{f} \in \mathbb{R}^d$ is the true image. Notice that the above formula uses values of $\mathbf{f}$ beyond the boundaries (indices smaller than 1 and larger than n), but these values are determined by the boundary conditions. For example, consider a one-dimensional image of length three: $\mathbf{f} = (f_1, f_2, f_3)^T$, and let the PSF array be $\mathbf{p} = (p_1, p_2, p_3)^T$ with $c = 2$. Then the blurred image $\mathbf{g}$ depends on the true image $\mathbf{f}$ via the relation $\mathbf{g} = \mathbf{S}\mathbf{f}$, where

$$\mathbf{S} = \begin{pmatrix} p_2 & p_1 & 0 \\ p_3 & p_2 & p_1 \\ 0 & p_3 & p_2 \end{pmatrix}, \begin{pmatrix} p_2 & p_1 & p_3 \\ p_3 & p_2 & p_1 \\ p_1 & p_3 & p_2 \end{pmatrix}, \begin{pmatrix} p_2 + p_3 & p_1 & 0 \\ p_3 & p_2 & p_1 \\ 0 & p_3 & p_2 + p_1 \end{pmatrix}$$

for zero, periodic, and reflexive boundary conditions, respectively. Note that the above three matrices have different structures (Toeplitz, circulant, and Toeplitz-plus-

Hankel). We now discuss four SD structures arising from one- and two-dimensional problems with either periodic or reflexive boundary conditions:[2]

1. [illegible] [10]. For one-dimensional images with periodic boundary conditions, the structure of the model matrix is circulant, i.e., has the form

$$\mathbf{S} = \begin{pmatrix} s_1 & s_2 & \cdots & s_n \\ s_n & s_1 & \cdots & s_{n-1} \\ \vdots & \vdots & & \vdots \\ s_2 & s_3 & \cdots & s_1 \end{pmatrix}.$$

All $n \times n$ circulant matrices are diagonalizable by the unitary discrete Fourier transform (DFT) matrix $\mathbf{F}_n$ given by

$$\mathbf{F}_n = \left(\frac{1}{\sqrt{n}} \omega^{(j-1)(k-1)} \right)_{j,k=1}^{n},$$

where $\omega = e^{\frac{2\pi \mathbf{i}}{n}}$. Multiplications of the DFT matrix $\mathbf{F}_n$ with vectors, as well as eigenvalue computation of circulant matrices, can be done very efficiently by using the fast Fourier transform (FFT) with a complexity of $O(n \log n)$.

2. [illegible] [2]. For two-dimensional images of size $m \times n$ with periodic boundary conditions, the model matrix has a block circulant matrix with circulant blocks (BCCB) structure:

$$\mathbf{S} = \begin{pmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \dots & \mathbf{C}_n \\ \mathbf{C}_n & \mathbf{C}_1 & \dots & \mathbf{C}_{n-1} \\ \vdots & \vdots & & \vdots \\ \mathbf{C}_2 & \mathbf{C}_3 & \dots & \mathbf{C}_1 \end{pmatrix},$$

where $\mathbf{C}_1, \dots, \mathbf{C}_n$ are $m \times m$ circulant matrices. All BCCB matrices of the above size are diagonalizable by the unitary two-dimensional DFT matrix $\mathbf{F}_n \otimes \mathbf{F}_m$. As in the circulant case, computations with BCCB matrices can be performed by using the FFT.

3. [illegible] [30]. For one-dimensional images with reflexive boundary conditions and symmetric PSF, the matrix $\mathbf{S}$ has a Toeplitz-plus-Hankel structure of the form [30]

$$T(\mathbf{s}) + H(\mathbf{s}),$$

where, for a given vector $\mathbf{s} = (s_1, \dots, s_n)^T \in \mathbb{R}^n$, $T(\mathbf{s})$ is the symmetric Toeplitz matrix whose first column is $\mathbf{s}$ and $H(\mathbf{s})$ is the Hankel matrix whose first and last columns are $(s_1, s_2, \dots, s_n, 0)^T$ and $(0, s_n, \dots, s_2, s_1)^T$, respectively. All Toeplitz-plus-Hankel matrices of the above form are diagonalizable by the orthogonal discrete cosine transform (DCT) matrix $\mathbf{C}_n$ given by

$$\mathbf{C}_n = \left(\sqrt{(2 - \delta_{k1})/n} \cos \frac{\pi(2j-1)(k-1)}{2n} \right)_{j,k=1}^{n},$$

where, for two indices i and j, δ_{ij} denotes the Kronecker sign. Multiplications of the DCT matrix $\mathbf{C}_n$ with vectors, as well as eigenvalue computation

[2] We do not consider in this paper the zero boundary condition as it does not lead to an SD structure.

of circulant matrices, can be done very efficiently by using the fast cosine transform (FCT) with a complexity of $O(n \log n)$.

4. BTTB+BTHB+BHTB+BHHB [15, 30]. For two-dimensional images of size $m \times n$ with reflexive boundary conditions and a symmetric PSF, the matrix $\mathbf{S}$ is a sum of a BTTB (block Toeplitz with Toeplitz blocks), BTHB (block Toeplitz with Hankel blocks), BHTB (block Hankel with Toeplitz blocks), and BHHB (block Hankel with Hankel blocks) matrices. All matrices of this form are diagonalizable by the orthogonal two-dimensional DCT matrix $\mathbf{C}_n \otimes \mathbf{C}_m$. We note that the symmetry condition does occur in practice, for example, the Gaussian model for atmospheric turbulence blur, out-of-focus blurs, and certain classes of Moffat blurs [15].

We have thus described four SD structures arising from one- and two-dimensional deblurring problems. The first two classes correspond to $\mathbb{F} = \mathbb{C}$ (since the DFT matrix is complex-valued), and the last two classes correspond to $\mathbb{F} = \mathbb{R}$. Coming back to the RSTLS problem, we note that it is very natural to assume that the boundary conditions also apply to the regularization operator, and we can thus assume that $\mathbf{L} \in \mathcal{L}_{\mathbf{Q}}$.

3. Decomposition of the RSTLS problem for SD structures. We begin by showing that the RSTLS problem (2.1) with an SD structure can be decomposed into n one-dimensional minimization problems.

THEOREM 3.1. (2.1) $m = n$ $\mathcal{L} = \mathcal{L}_{\mathbf{Q}}$ (2.2) $\mathbf{Q} \in \mathbb{F}^{n \times n}$ $\mathbf{A}, \mathbf{L} \in \mathcal{L}_{\mathbf{Q}}$ $\boldsymbol{\alpha}, \mathbf{l}$ $\mathbf{A}$ $\mathbf{L}$

$$\mathbf{QAQ}^* = \operatorname{diag}(\boldsymbol{\alpha}), \quad \mathbf{QLQ}^* = \operatorname{diag}(\mathbf{l}). \tag{3.1}$$

$\mathbf{x} = \mathbf{Q}^*\hat{\mathbf{x}}$ $i = 1, \ldots, n$ $\hat{\mathbf{x}}$ $\hat{x}_i$

$$\min_{\hat{x}_i} \left\{ \frac{|\alpha_i \hat{x}_i - \hat{b}_i|^2}{1 + |\hat{x}_i|^2} + \rho |l_i|^2 |\hat{x}_i|^2 \right\}, \tag{3.2}$$

$\hat{\mathbf{b}} = \mathbf{Qb}$ $\mathbf{E}$

$$\mathbf{E} = \mathbf{Q}^* \operatorname{diag}(\mathbf{r}) \mathbf{Q}, \tag{3.3}$$

$$r_i = -\frac{\overline{\hat{x}_i}(\alpha_i \hat{x}_i - \hat{b}_i)}{1 + |\hat{x}_i|^2}. \tag{3.4}$$

. By using the relation $\mathbf{w} = (\mathbf{A} + \mathbf{E})\mathbf{x} - \mathbf{b}$, we can rewrite (2.1) as the following problem in the variables $\mathbf{E}$ and $\mathbf{x}$:

$$\min_{\mathbf{E},\mathbf{x}} \{ \|\mathbf{E}\|^2 + \|(\mathbf{A} + \mathbf{E})\mathbf{x} - \mathbf{b}\|^2 + \rho \|\mathbf{Lx}\|^2 : \mathbf{E} \in \mathcal{L}_{\mathbf{Q}}, \mathbf{x} \in \mathbb{F}^n \},$$

which, by the unitarity property of $\mathbf{Q}$, is the same as

$$\min_{\mathbf{E},\mathbf{x}} \{ \|\mathbf{QEQ}^*\|^2 + \|\mathbf{Q}(\mathbf{A} + \mathbf{E})\mathbf{Q}^*\mathbf{Qx} - \mathbf{Qb}\|^2 + \rho \|\mathbf{QLQ}^*\mathbf{Qx}\|^2 : \mathbf{E} \in \mathcal{L}_{\mathbf{Q}}, \mathbf{x} \in \mathbb{F}^n \}. \tag{3.5}$$

Since $\mathbf{E} \in \mathcal{L}_{\mathbf{Q}}$, we can make the change of variables $\mathbf{QEQ}^* = \mathrm{diag}(\mathbf{r})$, where $\mathbf{r} \in \mathbb{F}^n$ is an unknown variables vector. By combining this with (3.1) we conclude that (3.5) can be reformulated as

$$\min_{\mathbf{r},\hat{\mathbf{x}}} \{\|\mathrm{diag}(\mathbf{r})\|^2 + \|\mathrm{diag}(\boldsymbol{\alpha} + \mathbf{r})\hat{\mathbf{x}} - \hat{\mathbf{b}}\|^2 + \rho\|\mathrm{diag}(\mathbf{l})\hat{\mathbf{x}}\|^2 : \mathbf{r}, \hat{\mathbf{x}} \in \mathbb{F}^n\},$$

where $\hat{\mathbf{x}} = \mathbf{Q}\mathbf{x}$, and more explicitly as

$$\min_{\mathbf{r},\hat{\mathbf{x}}} \left\{ \sum_{i=1}^{n} (|r_i|^2 + |(\alpha_i + r_i)\hat{x}_i - \hat{b}_i|^2 + \rho|l_i|^2|\hat{x}_i|^2) : \mathbf{r}, \hat{\mathbf{x}} \in \mathbb{F}^n \right\}.$$

The above optimization problem is separable with respect to the pairs of variables

$$(r_1, \hat{x}_1), (r_2, \hat{x}_2), \ldots, (r_n, \hat{x}_n),$$

implying that, for every i, the optimal $(r_i, \hat{x}_i)$ is the solution to the two-dimensional problem

$$\min_{r_i,\hat{x}_i} \left\{ |r_i|^2 + |(\alpha_i + r_i)\hat{x}_i - \hat{b}_i|^2 + \rho|l_i|^2|\hat{x}_i|^2 : r_i, \hat{x}_i \in \mathbb{F} \right\}. \tag{3.6}$$

Next, we fix $\hat{x}_i$ and minimize with respect to r_i. The result is

$$r_i = -\frac{\overline{\hat{x}_i}(\alpha_i\hat{x}_i - \hat{b}_i)}{1 + |\hat{x}_i|^2}.$$

By substituting the above expression back into the objective function of (3.6) with some simple algebraic manipulations, we arrive at the following equivalent problem in the single variable $\hat{x}_i$:

$$\min_{\hat{x}_i} \left\{ \frac{|\alpha_i\hat{x}_i - \hat{b}_i|^2}{1 + |\hat{x}_i|^2} + \rho|l_i|^2|\hat{x}_i|^2 \right\},$$

establishing the result. □

4. Solution and analysis of the RSTLS problem for SD structures. In this section we study the one-dimensional (1D) problems (3.2) arising in the decomposition of the RSTLS problem. We show in section 4.1 that, although these problems are not unimodal,[3] they can be transformed into (strictly) unimodal problems and consequently solved efficiently and globally. This is especially crucial in image deblurring applications in which there are hundreds of thousands or even millions of 1D problems to be solved. Based on the uniqueness and attainment properties of the 1D problems, corresponding conditions for the RSTLS problem are established in section 4.2.

4.1. Solution of the single-variable problem. Our goal in this section is to analyze the one-dimensional problem (3.2) and to devise an efficient solution method for solving it. Consider the problem

$$\min_{x\in\mathbb{F}} \left\{ f(x) = \frac{|ax - b|^2}{1 + |x|^2} + |c|^2|x|^2 \right\}, \tag{4.1}$$

[3] A function $f : I \to \mathbb{R}$, $I \subseteq \mathbb{R}$ being a closed interval, is (strictly) unimodal if it has a unique local minimizer on I and is (strictly) decreasing from the left boundary of the interval to this unique minimum and (strictly) increasing from the minimum to the right boundary of the interval.

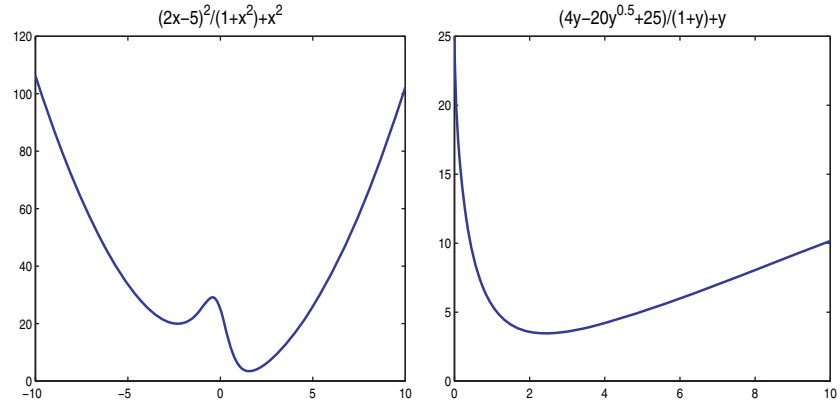


FIG. 1. *The objective function of problems* (4.5) *(left) and* (4.6) *(right).*

where $a, b, c \in \mathbb{F}$. If $c \neq 0$, then the objective function is coercive, and consequently its minimum is attained. The objective function of (4.1) is not unimodal (cf. Figure 1) and thus finding its global minimum efficiently is in principle a hard task. We will show in the next result that it can be solved via the minimization problem

$$\min_{y \geq 0} \left\{ g(y) \equiv \frac{|a|^2 y - 2|ab|\sqrt{y} + |b|^2}{1+y} + |c|^2 y \right\} \tag{4.2}$$

in the real nonnegative variable y. Before stating the result we briefly recall that for a real number $x \in \mathbb{R}$ the sign function is defined by

$$\operatorname{sgn}(x) \equiv \begin{cases} 1 & x > 0, \\ 0 & x = 0, \\ -1 & x < 0, \end{cases}$$

and for a complex number $z \in \mathbb{C}$ the sign function is given by

$$\operatorname{sgn}(z) \equiv \begin{cases} \frac{z}{|z|} & z \neq 0, \\ 0 & z = 0. \end{cases}$$

LEMMA 4.1 (equivalence of problems (4.1) and (4.2)). [illegible] (4.1) [illegible] $a, b, c \in \mathbb{F}$ [illegible]

(i) [illegible] $ab \neq 0$ [illegible] $\tilde{y}$ [illegible] (4.2) [illegible] $\tilde{x} =$ [illegible] $(\bar{a}b)\sqrt{\tilde{y}}$ [illegible] (4.1)

(ii) [illegible] $ab = 0$ [illegible] $\tilde{y}$ [illegible] (4.2) [illegible] $\tilde{x} = z\sqrt{\tilde{y}}$ [illegible] (4.1) [illegible] $z \in \mathbb{F}$ [illegible] $|z| = 1$ [illegible]. Let $\tilde{x}$ be an optimal solution of (4.1). Then by the optimality of $\tilde{x}$ we have

$$f(\tilde{x}) \leq f(z\tilde{x}) \text{ for every } z \in \mathbb{F} \text{ satisfying } |z| = 1,$$

which is the same as

$$\frac{|a\tilde{x} - b|^2}{1 + |\tilde{x}|^2} + |c|^2 |\tilde{x}|^2 \leq \frac{|a(z\tilde{x}) - b|^2}{1 + |z\tilde{x}|^2} + |c|^2 |z\tilde{x}|^2.$$

The latter inequality reduces to

$$\Re((1 - z)a\bar{b}\tilde{x}) \geq 0. \tag{4.3}$$

We will now show that $a\bar{b}\tilde{x}$ is a nonnegative real number. This is obviously true if $\tilde{x} = 0$. Otherwise, we split the analysis into two cases:

I. If $ab \neq 0$, then substituting

$$z = \frac{\overline{a\bar{b}\tilde{x}}}{|a\bar{b}\tilde{x}|}$$

into (4.3) yields

$$\Re(a\bar{b}\tilde{x}) \geq |a\bar{b}\tilde{x}|,$$

implying that $a\bar{b}\tilde{x}$ is a nonnegative real number and, in particular, that $\operatorname{sgn}(\tilde{x}) = \operatorname{sgn}(\bar{a}b)$.

II. If $ab = 0$, the function f satisfies $f(zx) = f(x)$ for every $x, z \in \mathbb{F}$ such that $|z| = 1$ and thus $z\tilde{x}$ is also an optimal solution for every z satisfying $|z| = 1$.

A conclusion from the above two cases is that if the minimum of (4.1) is attained at a nonzero solution, then there must be at least one optimal solution $\tilde{x}$ for which $\operatorname{sgn}(\tilde{x}) = \operatorname{sgn}(\bar{a}b)$; consequently, we can make the change of variables $x = \operatorname{sgn}(\bar{a}b)\sqrt{y}$ which transforms problem (4.1) into (4.2). □

4.1. Consider problem (4.1) with $\mathbb{F} = \mathbb{C}$ but with real data, i.e., $a, b, c \in \mathbb{R}$. Then a direct consequence of Lemma 4.1 is that if the optimal set of (4.1) is nonempty, then there must exist at least one real-valued optimal solution.

The following simple lemma establishes some key properties of problem (4.2). In particular, it is shown that problem (4.2) is strictly unimodal (in all interesting cases) and thus can be solved efficiently. This is in fact the main motivation for transforming problem (4.1) into (4.2).

LEMMA 4.2 (properties of problem (4.2)). (4.2) $a, b, c \in \mathbb{F}$

(i) $g(y)$ (4.2) [4] $[0, \infty)$

(ii) $c \neq 0$ $\tilde{y}$ (4.2) $\tilde{y} \leq \frac{|b|^2}{|c|^2}$

(iii) (4.2) $(a, c) \neq (0, 0)$

(iv) $(a, c) \neq (0, 0)$ $g(y)$ (4.2) $[0, \infty)$

(i) We need to show that the level set $\{y : g(y) \leq \alpha\}$ is convex. Indeed,

$$\{y \geq 0 : g(y) \leq \alpha\} = \{y \geq 0 : (|a|^2 + |c|^2 - \alpha)y - 2|ab|\sqrt{y} + |c|^2 y^2 + |b|^2 - \alpha \leq 0\}.$$

The latter is the zero level set of a convex function and hence convex.

(ii) Note that for $y \geq 0$

$$g(y) = \frac{(|a|\sqrt{y} - |b|)^2}{1 + y} + |c|^2 y \geq |c|^2 y.$$

Therefore, for $y > \frac{|b|^2}{|c|^2}$ we have

$$g(y) \geq |c|^2 y > |b|^2 = g(0),$$

showing that there are no optimal solutions for (4.2) larger than $\frac{|b|^2}{|c|^2}$.

[4] A function $f : I \to \mathbb{R}$ ($I \subseteq \mathbb{R}$ being an interval) is quasi-convex if all of its level sets $\{x \in I : f(x) \leq \alpha\}$ are convex.

(iii) First consider the case $(a, c) = (0, 0)$. Then $g(y) = |b|^2/(1 + y)$. Hence it follows either that g does not attain a minimum (if $b \neq 0$) or that the minimum (namely, all $y \geq 0$) is nonunique (if $b = 0$). Now consider the case $(a, c) \neq (0, 0)$. We split the analysis into two subcases.

I. If $c \neq 0$, then $\lim_{y\to\infty} g(y) = \infty$, implying the attainment of the minimum. To show the uniqueness of the minimum in this case, assume in contradiction that the optimal solution of (4.2) is not unique. Then since the optimal set is convex (by quasi convexity) we conclude that the optimal set is an interval $I \subseteq [0, \infty)$ with a nonempty interior. Denote the optimal value by f^*. Then

$$g(y) = f^* \text{ for every } y \in I,$$

which can be explicitly written as

$$(|a|^2 + |c|^2 - f^*)y - 2|ab|\sqrt{y} + |c|^2y^2 + |b|^2 - f^* = 0 \text{ for every } y \in I.$$

By making the change of variables $z = \sqrt{y}$, we obtain

$$(|a|^2 + |c|^2 - f^*)z^2 - 2|ab|z + |c|^2z^4 + |b|^2 - f^* = 0 \text{ for every } z \in J, \tag{4.4}$$

where $J = \{z : z^2 \in I\}$ is an interval with a nonempty interior. However, (4.4) is impossible since an univariate quartic equation has at most four roots and thus cannot have an infinite number of roots.

II. Suppose that $c = 0$. Then $a \neq 0$, and it is easy to see that g attains a unique minimum at $|b|^2/|a|^2$.

(iv) Since $(a, c) \neq (0, 0)$, we know from part (iii) that g attains a unique global minimum on the interval $[0, \infty)$. Hence it remains to show that it is strictly decreasing from the origin to this minimum and strictly increasing when we go from this minimum to plus infinity. Suppose that this is not true. Then the function g must have a stationary point in $(0, \infty)$ which is different from the unique minimum. However, we will show that, for any $y > 0$ such that $g'(y) = 0$, we automatically have $g''(y) > 0$; hence this stationary point y is at least a local minimum and, therefore, must be equal to the unique minimum of g on the interval $[0, \infty)$. Elementary differentiation gives

$$g'(y) = \frac{(|a|\sqrt{y} - |b|)(|a|\frac{1}{\sqrt{y}} + |b|)}{(1 + y)^2} + |c|^2.$$

Now let $\tilde{y} > 0$ be such that $g'(\tilde{y}) = 0$. Then

$$\begin{aligned} g''(\tilde{y}) &= -\frac{2}{(1+\tilde{y})^3}(|a|\sqrt{\tilde{y}} - |b|)\left(|a|\frac{1}{\sqrt{\tilde{y}}} + |b|\right) + \frac{|a||b|}{2(1+\tilde{y})^2\sqrt{\tilde{y}}}\left(1 + \frac{1}{\tilde{y}}\right) \\ &\overset{g'(\tilde{y})=0}{=} \frac{2|c|^2}{1+\tilde{y}} + \frac{|a||b|}{2(1+\tilde{y})^2\sqrt{\tilde{y}}}\left(1 + \frac{1}{\tilde{y}}\right) > 0, \end{aligned}$$

where the positivity of the last expression comes from the fact that $(a, c) \neq (0, 0)$; hence this last expression can be equal to zero only if both $c = 0$ and $b = 0$, but then, by taking into account that $\tilde{y}$ is also a stationary point, we would obtain $a = 0$ as well, in contrast to $(a, c) \neq (0, 0)$. ∎

The most important property of the function g is its strict unimodality (as stated in Lemma 4.2 (iv)). The strict unimodality property implies that there are no nonglobal local minima and thus enables us to invoke efficient one-dimensional solvers for

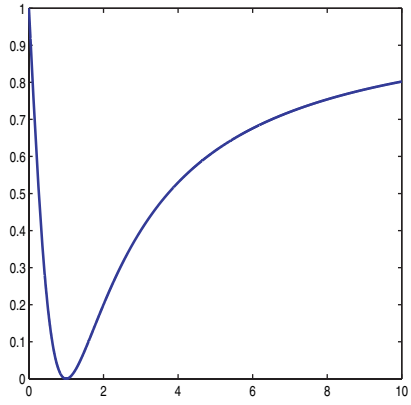


FIG. 2. *The function* $\frac{(\sqrt{y}-1)^2}{1+y}$ *from Remark* 4.2.

(strictly) unimodal functions that are guaranteed to converge to the global minimum. The following example illustrates this property.

1. Consider problem (4.1) with $\mathbb{F} = \mathbb{R}$, $a = 2, b = 5$, and $c = 1$. In this case, problems (4.1) and (4.2) are given by

$$\min_x \left\{ \frac{(2x-5)^2}{1+x^2} + x^2 \right\} \tag{4.5}$$

and

$$\min_{y \geq 0} \left\{ \frac{4y - 20\sqrt{y} + 25}{1+y} + y \right\}, \tag{4.6}$$

respectively. The plots of the two functions are given in Figure 1.

Clearly, the objective function in (4.5) is not unimodal and indeed possesses a nonglobal local minimizer. The global solution of (4.5) is $\tilde{x} = 1.5606$ (given in four-digit accuracy). The objective function in (4.6) is, as guaranteed by Lemma 4.1, an unimodal function. The global minimum is $\tilde{y} = 2.4354$, and the relation $\tilde{x} = \sqrt{\tilde{y}}$ holds.

4.2. A natural question here is whether g is even more than quasi-convex, namely, convex. The answer to this question is negative. For example, for $a = b = 1$ and $c = 0$, the function g is clearly nonconvex as can be seen from Figure 2. Note, however, that this figure also illustrates the quasi convexity of g.

Combining Lemmas 4.1 and 4.2, we are now able to state the basic properties of problem (4.1).

LEMMA 4.3 (uniqueness for problem (4.1)).
(4.1).
(i) $a \neq 0$
(ii) $a = 0, c \neq 0$ $|b| \leq |c|$

. We will split the analysis into four cases:

I. $a \neq 0$ and $b \neq 0$. By Lemma 4.2(iii), since $a \neq 0$, the optimal solution of (4.2) is uniquely attained. Moreover, since $ab \neq 0$, then by Lemma 4.1, there is a one-to-one correspondence between optimal solutions of (4.1) and (4.2) (via the relation $\tilde{x} = \operatorname{sgn}(\bar{a}b)\sqrt{\bar{y}}$), implying the uniqueness and attainment of the optimal solution of (4.1).

II. $a \neq 0$ and $b = 0$. The objective function of (4.2) in this case is strictly increasing, implying that the unique optimal solution of (4.2) is $\tilde{y} = 0$ and hence that the unique optimal solution of (4.1) is $\tilde{x} = 0$.

III. $a = 0$ and $b \neq 0$. By Lemma 4.2(iii), to guarantee the uniqueness and attainment of the optimal solution of (4.2) we must further assume that $c \neq 0$. The solution of (4.1) is unique if and only if the optimal solution $\tilde{y}$ of (4.2) is zero (otherwise, $z\sqrt{\tilde{y}}$ will be an optimal solution of (4.1) for every z satisfying $|z| = 1$). By the unimodality of g, the optimal solution is 0 if and only if $g'(0) \geq 0$, which is equivalent to $|b| \leq |c|$.

IV. $a = 0$ and $b = 0$. Again, as in the previous case, we further assume that $c \neq 0$. Here it is evident that the unique optimal solution is $\tilde{x} = 0$.

By combining the four cases we obtain the result. ∎

4.2. Uniqueness and attainment of the RSTLS solution. The result in section 4.1 collectively can be summed up in the following result.

THEOREM 4.1. [illegible] (2.1) [illegible] $m = n$ [illegible] $\mathcal{L} = \mathcal{L}_{\mathbf{Q}}$ ([illegible] (2.2)) [illegible] $\mathbf{Q} \in \mathbb{F}^{n\times n}$ [illegible] $\hat{\mathbf{b}} = \mathbf{Q}^*\mathbf{b}$ [illegible] $\mathbf{A}, \mathbf{L} \in \mathcal{L}_{\mathbf{Q}}$ [illegible] $\boldsymbol{\alpha}, \mathbf{l}$ [illegible] $\mathbf{A}$ [illegible] $\mathbf{L}$ [illegible]

$$\mathbf{Q}^*\mathbf{A}\mathbf{Q} = \operatorname{diag}(\boldsymbol{\alpha}), \quad \mathbf{Q}^*\mathbf{L}\mathbf{Q} = \operatorname{diag}(\mathbf{l}). \tag{4.7}$$

[illegible] $i = 1, \ldots, n$ [illegible]

(i) $\alpha_i \neq 0$

(ii) $\alpha_i = 0, l_i \neq 0$ [illegible] $|\hat{b}_i| \leq \sqrt{\rho}|l_i|$

[illegible]. Note that the optimal $\mathbf{E}$ is uniquely defined via the optimal $\mathbf{x}$ by (3.3) and (3.4). Therefore, the uniqueness and/or attainment properties of the optimal solution of (2.1) amount to uniqueness and/or attainment of the single-variable problems (3.2), which combined with Lemma 4.3 establishes the result. ∎

Theorem 4.1 provides conditions for the optimal solution of the RSTLS problem to be uniquely attained. Based on this, we can derive a simpler condition:

$$\operatorname{Null}(\mathbf{A}) \cap \operatorname{Null}(\mathbf{L}) = \{\mathbf{0}\}, \tag{4.8}$$

which is sufficient for attainment of the optimal solution and necessary for the unique attainment of the optimal solution, as shown in the following theorem.

THEOREM 4.2. [illegible] 4.1 [illegible]

(i) [illegible] (2.1) [illegible] (4.8) [illegible]

(ii) [illegible] (4.8) [illegible] (2.1) [illegible]

(iii) [illegible] $\mathbf{A}$ [illegible] (2.1) [illegible]

[illegible]. (i) Note that by Theorem 4.1 a necessary condition for the optimal solution of (2.1) to be uniquely attained is that $|\alpha_i|^2 + |l_i|^2 \neq 0$ for every i, that is, α_i and l_i are not both zero for any given i. The eigenvalues of the matrix $\mathbf{A}^*\mathbf{A} + \mathbf{L}^*\mathbf{L}$ are exactly $|\alpha_i|^2 + |l_i|^2$, implying that $\mathbf{A}^*\mathbf{A} + \mathbf{L}^*\mathbf{L}$ is nonsingular; therefore,

$$\operatorname{Null}(\mathbf{A}) \cap \operatorname{Null}(\mathbf{L}) = \operatorname{Null}(\mathbf{A}^*\mathbf{A} + \mathbf{L}^*\mathbf{L}) = \{\mathbf{0}\}.$$

(ii) Assume that condition (4.8) holds. By Theorem 3.1, it is enough to show that for every $i = 1, \ldots, n$ the one-dimensional problem (3.2) has at least one optimal solution. Now by Lemma 4.1 it is sufficient to establish the attainment of the solution

of

$$\min_{y\geq 0}\left\{\frac{|\alpha_i|^2 y - 2|\alpha_i\hat{b}_i|\sqrt{y}+|\hat{b}_i|^2}{1+y}+\rho|l_i|^2 y\right\} \tag{4.9}$$

for every $i=1,\dots,n$, where $\alpha_i,\hat{b}_i$, and l_i are defined in the premise of Theorem 3.1. By Lemma 4.2(iii), this is guaranteed if $(\alpha_i,l_i)\neq(0,0)$ for every i, which, as shown in the proof of (i), is equivalent to condition (4.8).

(iii) It follows from the nonsingularity of $\mathbf{A}$ that all of its eigenvalues are nonzero, which, by Theorem 4.1, implies that the solution of (2.1) is uniquely attained. □

The following example shows by suitable counterexamples that the assumptions used in Theorem 4.2 are sufficient, but not necessary, for the corresponding statements to be true.

2. (i) Consider problem (2.1) with $n=m=1, A=(0), L=(1), b=(2), \rho=1$, and $\mathbb{F}=\mathbb{R}$. Then condition (4.8) holds, but problem (2.1) has the two solutions $(E,x)=(1,1)$ and $(E,x)=(-1,-1)$. This shows that the unique attainment of a solution of problem (2.1) is sufficient for condition (4.8) to hold but not necessary.

(ii) Consider problem (2.1) with $n=m=1, A=(0), L=(0), b=(0), \rho=1$, and $\mathbb{F}=\mathbb{R}$. Then every vector (E,x), with $E=0$ and $x\in\mathbb{R}$ arbitrary, is a solution of problem (2.1), although condition (4.8) does not hold. Hence this condition is sufficient for problem (2.1) to have a nonempty solution set but not necessary.

It is interesting to compare the above conditions to the corresponding attainment/uniqueness conditions for the regularized least squares problem:

$$\text{(RLS):}\quad \min \|\mathbf{Ax}-\mathbf{b}\|^2+\rho\|\mathbf{Lx}\|^2.$$

The optimal solution of (RLS), as opposed to the solution of the RSTLS problem, is always attained; it is unique if and only if condition (4.8) holds. This is in contrast to the RSTLS problem where condition (4.8) is only a necessary condition for unique attainment of the solution.

5. The RSTLS problem with circulant structure. The RSTLS problem (2.1) with $\mathcal{L}=\mathcal{L}_{\mathbf{F}_n}$ ($\mathbf{F}_n$ being the $n\times n$ DFT matrix) corresponds to problems with circulant-structured matrices. Here the underlying number field is $\mathbb{F}=\mathbb{C}$ since the matrix $\mathbf{F}_n$ is complex-valued. However, in many applications the data $\mathbf{A},\mathbf{b}$, and $\mathbf{L}$ are real-valued. The main result in this section is that if the optimal set of the RSTLS problem is nonempty, then there exists at least one real-valued optimal solution. Therefore, there is no drawback in analyzing the RSTLS problem over the complex field even when the data are real-valued.

THEOREM 5.1. [illegible] $\mathbb{F}=\mathbb{C}, \mathcal{L}=\mathcal{L}_{\mathbf{F}_n}$ [illegible] $\mathbf{F}_n$ [illegible] $n\times n$ [illegible] $\mathbf{A}$, $\mathbf{b}$ [illegible] $\mathbf{L}$ [illegible] $\mathbf{A}\in\mathbb{R}^{n\times n}, \mathbf{b}\in\mathbb{R}^n$, [illegible] $\mathbf{L}\in\mathbb{R}^{n\times n}$ [illegible] (RSTLS) [illegible]. We will require the following notation:

$$\mathcal{A}=\{\mathbf{z}\in\mathbb{C}^n : z_1\in\mathbb{R}, z_{j+1}=\overline{z_{n+1-j}} \text{ for every } j=1,\dots,n-1\}.$$

To simplify the notation we omit the subscript in the $n\times n$ DFT matrix and denote it by $\mathbf{F}$ rather than by $\mathbf{F}_n$. The proof is based on the following three claims:

(i) Let $\mathbf{w}=\mathbf{Fv}$ for some $\mathbf{v}\in\mathbb{R}^n$. Then $\mathbf{w}\in\mathcal{A}$.

(ii) Let $\boldsymbol{\alpha}$ be the vector of eigenvalues of a real-valued circulant matrix $\mathbf{A}$. Then $\boldsymbol{\alpha} \in \mathcal{A}$.

(iii) Let $\mathbf{z} \in \mathcal{A}$. Then $\mathbf{F}^*\mathbf{z} \in \mathbb{R}^n$.

Proof. (i). First,

$$w_1 = (\mathbf{F}\mathbf{v})_1 = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} v_i,$$

proving that $w_1 \in \mathbb{R}$. Next, for every $j = 1, \ldots, n-1$ we have

$$w_{j+1} = (\mathbf{F}\mathbf{v})_{j+1} = \sum_{i=1}^{n} F_{j+1,i} v_i = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \omega^{j(i-1)} v_i. \tag{5.1}$$

On the other hand,

$$\begin{aligned} \overline{w_{n+1-j}} = \overline{(\mathbf{F}\mathbf{v})_{n+1-j}} &= \sum_{i=1}^{n} \overline{F_{n+1-j,i}} v_i = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \overline{\omega^{(n-j)(i-1)}} v_i \\ &= \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \omega^{j(i-1)} v_i \overset{(5.1)}{=} w_{j+1}. \end{aligned}$$

Proof. (ii). Let $(s_1, s_2, \ldots, s_n)$ be the first row of $\mathbf{A}$. The jth eigenvalue of the circulant matrix $\mathbf{A}$ is given by $\alpha_j = \sum_{i=1}^{n} \omega^{(i-1)(j-1)} s_i$. Then

$$\alpha_1 = \sum_{i=1}^{n} s_i \in \mathbb{R}$$

and

$$\overline{\alpha_{n+1-j}} = \overline{\sum_{i=1}^{n} \omega^{(i-1)(n-j)} s_i} = \sum_{i=1}^{n} \omega^{(i-1)j} s_i = \alpha_{j+1}$$

for every $j = 1, \ldots, n-1$. Thus, $\boldsymbol{\alpha} \in \mathcal{A}$.

Proof. (iii). For every $i = 1, 2, \ldots, n$:

$$\begin{aligned} \sqrt{n}(\mathbf{F}^*\mathbf{w})_i &= \sqrt{n} \sum_{j=1}^{n} \overline{F_{j,i}} w_j = \sum_{j=1}^{n} \omega^{-(i-1)(j-1)} w_j \\ &\overset{\mathbf{w} \in \mathcal{A}}{=} w_1 + \sum_{j=2}^{n} \omega^{-(i-1)(j-1)} \overline{w_{n+2-j}} = w_1 + \sum_{j=2}^{n} \omega^{(i-1)(n+1-j)} \overline{w_{n+2-j}} \\ &\overset{k \leftarrow n+2-j}{=} w_1 + \sum_{k=2}^{n} \omega^{(i-1)(k-1)} \overline{w_k} = \sqrt{n} \sum_{k=1}^{n} F_{k,i} \overline{w_k} \\ &= \overline{\sqrt{n} \sum_{k=1}^{n} \overline{F_{k,i}} w_k} = \sqrt{n}\, \overline{(\mathbf{F}^*\mathbf{w})_i}. \end{aligned}$$

By Theorem 3.1, an optimal solution of the RSTLS problem is given by $\mathbf{x} = \mathbf{F}^*\hat{\mathbf{x}}$, where $\hat{x}_i$, the ith component of $\hat{\mathbf{x}}$, is an optimal solution of (3.2). Recall that $\hat{\mathbf{b}} = \mathbf{F}\mathbf{b}$ for real-valued $\mathbf{b}$ and that $\boldsymbol{\alpha}$ and $\mathbf{l}$ are the eigenvalues vectors of the real-valued circulant matrices $\mathbf{A}$ and $\mathbf{L}$, respectively. Therefore, by properties (i) and (ii), $\hat{\mathbf{b}}, \boldsymbol{\alpha}, \mathbf{l} \in$

$\mathcal{A}$. Hence, $\hat{x}_1$ is the solution of (3.2) with $i = 1$ and with real data, which by Remark 4.1 implies that $\hat{x}_1$ is real. Moreover, for every $j = 1, \ldots, n-1$, $\hat{x}_j$ and $\hat{x}_{n+1-j}$ are the optimal solutions of

$$\min_{\hat{x}_{j+1}} \left\{ \frac{|\alpha_{j+1}\hat{x}_{j+1} - \hat{b}_{j+1}|^2}{1 + |\hat{x}_{j+1}|^2} + \rho |l_j|^2 |\hat{x}_j|^2 \right\},$$

$$\min_{\hat{x}_{n+1-j}} \left\{ \frac{|\alpha_{j+1}\overline{\hat{x}_{n+1-j}} - \hat{b}_{j+1}|^2}{1 + |\overline{\hat{x}_{n+1-j}}|^2} + \rho |l_{j+1}|^2 |\overline{\hat{x}_{n+1-j}}|^2 \right\},$$

respectively. Therefore, we can always choose the optimal solutions of these problems to satisfy $\hat{x}_{n+1-j} = \overline{\hat{x}_{j+1}}$. Thus, for the mentioned choice $\hat{\mathbf{x}} \in \mathcal{A}$ and by property (iii) this proves that $\mathbf{x} = \mathbf{F}^* \hat{\mathbf{x}}$ is real-valued. □

5.1. It can be shown by using the same methodology employed in the proof of Theorem 5.1 that there always exists a real-valued solution for the RSTLS problem with $\mathbf{Q} = \mathbf{F}_n \otimes \mathbf{F}_m$ (BCCB structure) whenever $\mathbf{A}, \mathbf{L}$, and $\mathbf{b}$ are real-valued.

The following two examples demonstrate the validity of Theorem 5.1.

3. Let $\mathbf{Q} = \mathbf{F}_3$ (3×3 circulant matrices) and

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}, \qquad \mathbf{L} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix}, \qquad \rho = 1.$$

Then

$$\boldsymbol{\alpha} = \operatorname{diag}(\mathbf{F}_3 \mathbf{A} \mathbf{F}_3^*) = \begin{pmatrix} 6 \\ -1.5 - 0.866025\mathbf{i} \\ -1.5 + 0.866025\mathbf{i} \end{pmatrix},$$

$$\hat{\mathbf{b}} = \mathbf{F}_3 \mathbf{b} = \begin{pmatrix} 8.660254 \\ -0.866025 + 0.5\mathbf{i} \\ -0.866025 - 0.5\mathbf{i} \end{pmatrix}, \qquad \mathbf{l} = \begin{pmatrix} 0 \\ 1.5 - 0.866025\mathbf{i} \\ 1.5 + 0.866025\mathbf{i} \end{pmatrix}.$$

The vector $\hat{\mathbf{x}}$ consisting of the optimal solutions the three arising optimization problems is

$$\hat{\mathbf{x}} = \begin{pmatrix} 1.443375 \\ 0.143941 - 0.249314\mathbf{i} \\ 0.143941 + 0.249314\mathbf{i} \end{pmatrix},$$

and the optimal solution

$$\mathbf{x} = \mathbf{F}_3^* \hat{\mathbf{x}} = \begin{pmatrix} 0.999543 \\ 0.999543 \\ 0.500913 \end{pmatrix}$$

is indeed real.

4. Consider the RSTLS problem with $\mathbf{Q} = \mathbf{F}_3$ (3×3 circulant matrices) and

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}, \qquad \mathbf{L} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix}, \qquad \rho = 1.$$

Then

$$\boldsymbol{\alpha} = \operatorname{diag}(\mathbf{F}_3\mathbf{A}\mathbf{F}_3^*) = \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix}, \qquad \hat{\mathbf{b}} = \mathbf{F}_3\mathbf{b} = \begin{pmatrix} 6.928203 \\ -1.732050 + \mathbf{i} \\ -1.732050 - \mathbf{i} \end{pmatrix}.$$

In this example the optimal solutions of the arising one-dimensional problems are not unique, and they consist of the collection of vectors $\hat{\mathbf{x}}$ of the form:

$$\hat{\mathbf{x}} = \begin{pmatrix} 2.309401 \\ 0.393319z_1 \\ 0.393319z_2 \end{pmatrix},$$

where z_1 and z_2 are complex numbers satisfying $|z_1| = |z_2| = 1$. Correspondingly, the set of optimal solutions of (RSTLS) consists of all vectors $\mathbf{F}_3^*\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ is of the above form and is thus equal to

$$\{(a + z_1 b + z_2 c, a + b z_1 \bar{\omega} + c z_2 \omega, a + b z_1 \omega + c z_2 \bar{\omega})^T : |z_1| = |z_2| = 1\},$$

where $a = 2.309401, b = 0.393319$, and $\omega = e^{\frac{2\pi\mathbf{i}}{3}}$. The above set certainly contains complex-valued optimal solutions, but, if we choose $z_1 = \bar{z}_2$, we obtain a subset of real-valued optimal solutions:

$$\left\{ \frac{1}{\sqrt{3}}(a + 2\cos(\theta)c, a + 2\cos(\theta + 2\pi/3)c, a + 2\cos(\theta - 2\pi/3)b)^T : 0 \le \theta \le 2\pi \right\}.$$

6. Solution of the CSTLS problem with SD structure. When the regularization is made by adding a constraint rather than by penalization, the problem becomes

$$\text{(CSTLS):} \quad \begin{array}{ll} \min\limits_{\mathbf{E},\mathbf{x}} & \|\mathbf{E}\|^2 + \|(\mathbf{A} + \mathbf{E})\mathbf{x} - \mathbf{b}\|^2 \\ \text{s.t.} & \|\mathbf{L}\mathbf{x}\|^2 \le \alpha, \\ & \mathbf{E} \in \mathcal{L}_{\mathbf{Q}}, \\ & \mathbf{x} \in \mathbb{F}^n, \end{array} \tag{6.1}$$

where $\alpha > 0$. We will show that the CSTLS problem can be solved by a sequence of RSTLS problems using a dual approach. We assume throughout this section that $\mathbf{A}$ is nonsingular. This assumption prevails in many image deblurring problems, although the matrix is often extremely ill conditioned.

The Lagrangian dual problem of (6.1) is given by

$$\max_{\lambda \ge 0} q(\lambda), \tag{6.2}$$

where

$$q(\lambda) = \begin{array}{ll} \min\limits_{\mathbf{E},\mathbf{x}} & \|\mathbf{E}\|^2 + \|(\mathbf{A} + \mathbf{E})\mathbf{x} - \mathbf{b}\|^2 + \lambda(\|\mathbf{L}\mathbf{x}\|^2 - \alpha) \\ \text{s.t.} & \mathbf{E} \in \mathcal{L}_{\mathbf{Q}}, \mathbf{x} \in \mathbb{F}^n. \end{array} \tag{6.3}$$

Therefore, evaluating a value of the dual objective function amounts to solving a single RSTLS problem which can be solved efficiently as shown in the previous sections. Since $\mathbf{A}$ is nonsingular, then by Theorem 4.2 (iii), the optimal solution of (6.3) is uniquely attained for all $\lambda \ge 0$, and we denote it by $(\mathbf{x}_\lambda, \mathbf{E}_\lambda)$. The function q has

several important properties which are summarized in Lemma 6.1 below. The differentiability property of q (part (ii) of Lemma 6.1), relies on the uniqueness property and on the following well known result [9, Proposition 6.1.1].

THEOREM 6.1. [illegible] f [illegible] g [illegible] X [illegible]

$$h(\lambda) \equiv \min_{\mathbf{x}\in X} \{f(\mathbf{x}) + \lambda g(\mathbf{x})\}, \quad \lambda \in [\lambda_1, \lambda_2],$$

[illegible] $\mathbf{x}_\lambda$ [illegible] $\lambda \in [\lambda_1, \lambda_2]$ [illegible] $\mathbf{x}_\lambda$ [illegible] h [illegible] $\lambda \in (\lambda_1, \lambda_2)$ [illegible] $h'(\lambda) = g(\mathbf{x}_\lambda)$

In our case the compactness assumption is not satisfied; however, this difficulty can be avoided. We will use the following notation:

$$\begin{aligned} s(\mathbf{x}, \mathbf{E}) &= \|\mathbf{E}\|^2 + \|(\mathbf{A} + \mathbf{E})\mathbf{x} - \mathbf{b}\|^2, \\ t(\mathbf{x}, \mathbf{E}) &= \|\mathbf{L}\mathbf{x}\|^2 - \alpha, \\ Y &= \{(\mathbf{x}, \mathbf{E}) : \mathbf{x} \in \mathbb{F}^n, \mathbf{E} \in \mathcal{L}_{\mathbf{Q}}\}. \end{aligned}$$

Then, in this notation, the CSTLS problem can be written as

$$\min_{\mathbf{x},\mathbf{E}} \{s(\mathbf{x}, \mathbf{E}) : t(\mathbf{x}, \mathbf{E}) \le 0, (\mathbf{x}, \mathbf{E}) \in Y\}. \tag{6.4}$$

LEMMA 6.1. [illegible] q [illegible] (6.3) [illegible]

(i) q [illegible] $[0, \infty)$

(ii) $q(\lambda)$ [illegible] $\lambda > 0$ [illegible] $q'(\lambda) = \|\mathbf{L}\mathbf{x}_\lambda\|^2 - \alpha$

(iii) $\lim_{\lambda\to\infty} q(\lambda) = -\infty$

[illegible] (i) $q(\lambda)$ is the pointwise minimum of functions which are linear in λ and hence concave.

(ii) Let $\tilde{\lambda} > 0$, and let $\lambda_2 > \lambda_1 > 0$ be two positive numbers for which $\tilde{\lambda} \in (\lambda_1, \lambda_2)$. The dual objective can be written as

$$q(\lambda) = \min\{s(\mathbf{x}, \mathbf{E}) + \lambda t(\mathbf{x}, \mathbf{E}) : (\mathbf{x}, \mathbf{E}) \in Y\}. \tag{6.5}$$

From the nonsingularity of $\mathbf{A}$ and Theorem 4.2(iii) it follows that there exists a unique minimizer to the above problem which we denote by $(\mathbf{x}_\lambda, \mathbf{E}_\lambda)$. By Theorem 3.1 it follows that $\mathbf{x}_\lambda = \mathbf{Q}^*\mathbf{y}^\lambda$, where the ith component of $\mathbf{y}^\lambda$, y_i^λ, is the solution to

$$\min_{y_i} \left\{ \frac{|\alpha_i y_i - \hat{b}_i|^2}{1 + |y_i|^2} + \rho\lambda |l_i|^2 |y_i|^2 \right\}.$$

If $l_i = 0$, then $y_i^\lambda = \frac{\hat{b}_i}{\alpha_i}$ ($\alpha_i \neq 0$ for every i as an eigenvalue of a nonsingular matrix). Otherwise,

$$\rho\lambda |l_i|^2 |y_i^\lambda|^2 \le \frac{|\alpha_i y_i^\lambda - \hat{b}_i|^2}{1 + |y_i^\lambda|^2} + \rho\lambda |l_i|^2 |y_i^\lambda|^2 \le \frac{|\alpha_i 0 - \hat{b}_i|^2}{1 + 0^2} + \rho\lambda |l_i|^2 0^2 = |\hat{b}_i|^2,$$

so that $|y_i^\lambda|^2 \le \frac{|\hat{b}_i|^2}{\rho\lambda |l_i|^2}$. Consequently, for every $\lambda \in [\lambda_1, \lambda_2]$,

$$|y_i^\lambda|^2 \le \begin{cases} \left|\frac{\hat{b}_i}{\alpha_i}\right|^2 & l_i = 0, \\ \frac{|\hat{b}_i|^2}{\rho\lambda_1 |l_i|^2} & l_i \neq 0. \end{cases}$$

Hence, $\mathbf{y}^\lambda$ is bounded for every $\lambda \in [\lambda_1, \lambda_2]$ showing that $\mathbf{x}_\lambda = \mathbf{Q}^*\mathbf{y}^\lambda$ is also bounded over $[\lambda_1, \lambda_2]$; that is, there exists $\beta > 0$ for which $\|\mathbf{x}_\lambda\| \le \beta, \lambda \in [\lambda_1, \lambda_2]$. Moreover, by the relation between the optimal $\mathbf{E}$ and the optimal $\mathbf{x}$ given by (3.3) and (3.4), it follows that $\mathbf{E}_\lambda$ is also bounded over $[\lambda_1, \lambda_2]$; namely, there exists $\gamma > 0$ for which $\|\mathbf{E}_\lambda\| \le \gamma$. The dual objective function can thus be written as

$$q(\lambda) = \min\{s(\mathbf{x}, \mathbf{E}) + \lambda t(\mathbf{x}, \mathbf{E}) : (\mathbf{x}, \mathbf{E}) \in \tilde{Y}\},$$

where

$$\tilde{Y} = \{(\mathbf{x}, \mathbf{E}) : \mathbf{x} \in \mathbb{F}^n, \mathbf{E} \in \mathcal{L}_\mathbf{Q}, \|\mathbf{x}\| \le \beta, \|\mathbf{E}\| \le \gamma\}$$

is a compact set. Therefore, by Theorem 6.1, q is differentiable over (λ_1, λ_2) and in particular at $\tilde{\lambda}$ and $q'(\tilde{\lambda}) = t(\mathbf{x}_{\tilde{\lambda}}, \mathbf{E}_{\tilde{\lambda}}) = \|\mathbf{L}\mathbf{x}_{\tilde{\lambda}}\|^2 - \alpha$.

(iii) Since $\mathbf{E} = \mathbf{0}, \mathbf{x} = \mathbf{0}$ is feasible for (6.3), we obtain

$$q(\lambda) \le \|\mathbf{b}\|^2 - \lambda\alpha,$$

establishing that $q(\lambda) \to -\infty$ as $\lambda \to \infty$. □

We will now show that, despite the nonconvexity of the CSTLS problem, strong duality holds.

THEOREM 6.2 (strong duality for CSTLS). [illegible] $\lambda^* > 0$ [illegible] (6.2) [illegible] $q(\lambda^*)$ [illegible] (6.1) [illegible] $(\mathbf{x}_{\lambda^*}, \mathbf{E}_{\lambda^*})$ [illegible] (6.1) [illegible]. Since $\lambda^* > 0$ is the optimal solution of (6.2) and q is differentiable by Lemma 4.2(ii), we have $\|\mathbf{L}\mathbf{x}_{\lambda^*}\|^2 - \alpha = q'(\lambda^*) = 0$. Therefore, $\mathbf{x}_{\lambda^*}$ is a feasible solution of the primal problem (6.1), and

$$q(\lambda^*) = s(\mathbf{x}_{\lambda^*}, \mathbf{E}_{\lambda^*}) + \lambda^*(\|\mathbf{L}\mathbf{x}_{\lambda^*}\|^2 - \alpha) = s(\mathbf{x}_{\lambda^*}, \mathbf{E}_{\lambda^*}),$$

which, from basic duality theory, implies that λ^* and $(\mathbf{x}_{\lambda^*}, \mathbf{E}_{\lambda^*})$ are the dual and primal optimal solutions, respectively. □

The optimal λ^* is a root of the nondecreasing function $q'(\lambda)$ and can thus be found via a simple bisection procedure.

7. Implementation and a numerical example.

7.1. Implementation. The core of the numerical method for solving the RSTLS problem is the solution of n single-variable problems of the form (4.1). Since the number of these 1D problems might be huge (for example, for a two-dimensional 1024×1024 image, there are more than one million problems), it is imperative to find the [illegible] solution of each of them. The method will produce an erroneous solution even if one of the 1D problems is not solved correctly.

From numerical considerations the algorithm is split into two phases. In the first phase, we find the optimal solution of (4.2) up to a moderate tolerance ε (in our experiments $\varepsilon = 10^{-4}$). That is, the output of the first phase is an interval $[\ell, u]$, with $u - \ell < \varepsilon$, in which the optimal solution of (4.2) is guaranteed to reside. The goal of the first phase is to find a "small enough" interval in which the global solution is guaranteed to reside. Since in the course of the change of variables $x = \operatorname{sgn}(\bar{a}b)\sqrt{y}$ the accuracy of the solution might be reduced from ε to $\sqrt{\varepsilon}$, a second phase is invoked in which we seek the global minimizer x^* of the problem

$$\min_x \left\{ \frac{|a|^2x^2 - 2|ab|x + |b|^2}{1 + x^2} + |c|^2x^2 \right\} \tag{7.1}$$

in the interval $[\sqrt{\ell}, \sqrt{u}]$ up to a tolerance ε^2. The interval $[\sqrt{\ell}, \sqrt{u}]$ is small enough so that for all practical purposes the function in (7.1) is unimodal over $[\sqrt{\ell}, \sqrt{u}]$ and the global optimal solution given by $\text{sign}(\bar{a}b)x^*$ is obtained. A detailed description of the algorithm follows.

ALGORITHM SOLVE1D(a, b, c).

input	$a, b, c \in \mathbb{C}$
output	x - an optimal solution of (4.1).
comments	1. It is assumed that a and c are not both zero.
	2. The functions f_1 and f_2 called in the solver are given by
	$f_1(x; a, b, c) = \frac{(\lvert a\rvert\sqrt{x}-\lvert b\rvert)^2}{1+x} + \lvert c\rvert^2 x,$
	$f_2(x; a, b, c) = \frac{(\lvert a\rvert x-\lvert b\rvert)^2}{1+x^2} + \lvert c\rvert^2 x^2.$

If c is equal to zero up to some tolerance, then the output of the algorithm is b/a; otherwise, the upper bound is chosen.

if $c < 10^{-8}$
  $x = \frac{b}{a}$
  stop
else
  $u = \left|\frac{b}{c}\right|^2$
end if
$\ell = 0$
$s = \text{sgn}\,(\bar{a}b)$
Phase I. Activating an unimodal solver on the function f_1
while $(u - \ell) > \varepsilon$
  $x^- = \frac{2}{3}\ell + \frac{1}{3}u$
  $x^+ = \frac{1}{3}\ell + \frac{2}{3}u$
  $f^+ = f_1(x^+; a, b, c)$
  $f^- = f_1(x^-; a, b, c)$
  if $f^- \le f^+$
    $u = x^+$
  else
    $\ell = x^-$
  end if
end while
Updating the lower and upper bounds.
$\ell = \sqrt{\ell}$
$u = \sqrt{u}$
Phase II. Activating an unimodal solver on the function f_2.
while $(u - \ell) > \varepsilon^2$
  $x^- = \frac{2}{3}\ell + \frac{1}{3}u$
  $x^+ = \frac{1}{3}\ell + \frac{2}{3}u$
  $f^+ = f_2(x^+; a, b, c)$
  $f^- = f_2(x^-; a, b, c)$
  if $f^- \le f^+$
    $u = x^+$
  else
    $\ell = x^-$
  end if

end while
$x = s\frac{x^+ + x^-}{2}$
stop

We note that in the MATLAB implementation the minimization of the n 1D problems is done simultaneously using MATLAB's vector operations. For the exact implementation please see the (small) RSTLS MATLAB package available at [38]. Given the 1D solver, the solution of the RSTLS problem (2.1) is obtained via the following procedure.

ALGORITHM RSTLS $(\mathbf{Q}, \mathbf{A}, \mathbf{L}, \rho)$.

input	$\mathbf{Q} \in \mathbb{F}^{n\times n}$ - a unitary matrix.
	$\mathbf{A}, \mathbf{L} \in \mathcal{L}_{\mathbf{Q}}, \mathbf{b} \in \mathbb{F}^n$.
	$\rho \in \mathbb{R}_{++}$.
output	The $\mathbf{x}$-part of the optimal solution of (2.1).

Step 1. $\hat{\mathbf{b}} = \mathbf{Q}\mathbf{b}$.
Step 2. Compute the eigenvalues vectors $\boldsymbol{\alpha}, \mathbf{l}$ of $\mathbf{A}$ and $\mathbf{L}$ defined by the relations (3.1).
Step 3. For each $i = 1, \ldots, n$ call algorithm SOLVE1D with input $\alpha_i, \hat{b}_i, c_i$ and obtain an output $\hat{x}_i$.
Step 4. $\mathbf{x} = \mathbf{Q}^*\hat{\mathbf{x}}$, where $\hat{\mathbf{x}} = (\hat{x}_i)_{i=1}^n$.

Based on the RSTLS algorithm, the constrained version, problem (CSTLS), is solved via a simple bisection algorithm applied to $q'(\lambda)$, where q is the dual function defined by (6.3). The bisection is over the logarithm of base 10 of the dual variable λ.

ALGORITHM CSTLS$(\mathbf{Q}, \mathbf{A}, \mathbf{L}, \alpha)$.

input	$\mathbf{Q} \in \mathbb{F}^{n\times n}$ - a unitary matrix.
	$\mathbf{A}, \mathbf{L} \in \mathcal{L}_{\mathbf{Q}}, \mathbf{b} \in \mathbb{F}^n$.
	$\alpha \in \mathbb{R}_{++}$.
output	The $\mathbf{x}$-part of the optimal solution of (6.1).

Step 1. $u = 2, \ell = -4$.
Step 2. **while** $(u - \ell) > 0.1$
$h = \frac{u+\ell}{2}$
call Algorithm RSTLS with input $\mathbf{Q}, \mathbf{A}, \mathbf{L}, 10^h$ and obtain an output $\tilde{\mathbf{x}}$
if $\|\mathbf{L}\tilde{\mathbf{x}}\|^2 < \alpha$
$u = h$
else
$\ell = h$
end if
end while
Step 3. $\mathbf{x} = \tilde{\mathbf{x}}$.

Note that the RSTLS and CSTLS algorithms use matrix-vector multiplications with the matrices $\mathbf{Q}$ and $\mathbf{Q}^*$ and require the computation of the eigenvalues of the matrices $\mathbf{A}$ and $\mathbf{L}$. When $\mathcal{L}_{\mathbf{Q}}$ is one of the four SD structures described in section 2.2 in the context of image deblurring, these operations can be efficiently performed by utilizing fast transforms: one- or two-dimensional FFT for periodic boundary conditions and one- or two-dimensional FCT for reflexive boundary conditions.

7.2. A numerical example. To demonstrate our approach we consider an image deblurring example. We start with the 512×512 Lena gray image (top left image of Figure 3) scaled so that all of the pixels are in the interval $[0, 1]$ and blur it with a Gaussian PSF of dimension 9×9 with standard deviation 6 implemented in the command psfGauss([9,9],6) from [15]; the values in the PSF range between 0.0095 and 0.0148. We assume that the blurring is not exactly known and that the observed PSF is a Gaussian PSF of dimension 9×9 with standard deviation 8. We then cut the margins by 20 rows and columns resulting in 492×492 and add a Gaussian white noise with standard deviation 10^{-3} (top right image of Figure 3). By assuming reflexive boundary conditions, the poor naive solution construction (i.e., $\mathbf{A}^{-1}\mathbf{b}$) is given in the left middle image of Figure 3. This poor quality of the naive solution is not surprising since the problem is extremely ill conditioned. In our experiments, the regularization matrix $\mathbf{L}$ represents a discretization of a differential operator corresponding to the PSF

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}.$$

The constrained least squares solution, that is, the solution of the problem

$$\min\{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 : \|\mathbf{L}\mathbf{x}\|^2 \leq \alpha\},$$

is presented in the right middle image. The CSTLS reconstructions under periodic and reflexive boundary conditions are the left and right bottom images, respectively. The parameter α is chosen as $1.2\|\mathbf{L}\mathbf{x}_{\text{true}}\|^2$. Clearly, the best reconstruction is provided by the CSTLS algorithm with reflexive boundary conditions. The artifacts in the CSTLS reconstruction with periodic boundary conditions are much more prominent. The relative error of the CSTLS reconstruction with reflexive boundary conditions, $\frac{\|\mathbf{x}_{\text{true}} - \mathbf{x}_{\text{CSTLS-R}}\|}{\|\mathbf{x}_{\text{true}}\|}$, is 0.0961, while for periodic boundary conditions the relative error is 0.1393. The constrained least squares solution gave the worst relative error: 0.15.

8. Conclusion and discussion. In this paper we have shown that the RSTLS problem for structures involving matrices which are simultaneously diagonalizable by a given unitary matrix can be efficiently and globally solved (as opposed to general structures). These SD structures appear in image deblurring problems with either reflexive or periodic boundary conditions. The solution method consists of first decomposing the problem into several real or complex one-dimensional problems which are not necessarily unimodal. In the described image deblurring examples, the decomposition is performed by using the FFT or the FCT. The one-dimensional problems are then globally solved by invoking an unimodal solver on a transformation of the problems. Numerical results demonstrate the effectiveness of the proposed approach.

Another type of boundary conditions are antireflective boundary conditions introduced in [35].[5] As stated in [35], antireflective boundary conditions further reduce the boundary artifacts. The reason is that zero Dirichlet and periodic boundary conditions introduce an artificial discontinuity at the border of the field of view; reflexive boundary conditions impose that the reflected image is globally continuous but introduce an artificial discontinuity of the first derivative, while antireflective boundary conditions using a central symmetry are able to maintain C^1 continuity in the case of signals and C^0 with normal derivative continuity for images.

[5] We thank an anonymous reviewer for referring us to the literature on this type of boundary conditions.

FIG. 3. *Deblurring of Lena.*

In analogy with the reflexive boundary conditions, matrix-vector operations, solution of linear systems, and eigenvalue computations in the antireflective setting can be done in $O(n \log n)$ real operations [4] (using the fast sine transform). It is also known

that for these types of boundary conditions and with symmetric PSFs the set of all possible matrices is simultaneously diagonalizable [3]. However, the diagonalizing matrix is not unitary. The unitary property is essential to the analysis introduced in the current paper. Specifically, the decomposition of the RSTLS problem described in Theorem 3.1 will not be valid if the diagonalization is via a nonunitary matrix. Therefore, it does not seem possible to analyze the RSTLS problem with antireflective boundary conditions within the setting of the paper. It is an open question whether it is possible to exploit the special properties of antireflective boundary conditions in order to construct an efficient method for solving the corresponding RSTLS problem.

Acknowledgment. We thank two anonymous referees for their helpful comments and suggestions which helped to improve the presentation of the paper.

REFERENCES

[1] T. J. Abatzoglou, J. M. Mendel, and G. A. Harada, *The constrained total least squares technique and its applications to harmonic supperresolution*, IEEE Trans. Signal Process., 39 (1991), pp. 1070–1087.

[2] H. Andrews and B. Hunt, *Digital Image Restoration*, Prentice–Hall, Englewood Cliffs, NJ, 1977.

[3] A. Arico', M. Donatelli, J. Nagy, and S. Serra-Capizzano, *The anti-reflective transform and regularization by filtering*, in Numerical Linear Algebra in Signals, Systems, and Control (NLASSC), Lecture Notes in Electrical Engineering, Springer-Verlag, New York, to appear.

[4] A. Arico', M. Donatelli, and S. Serra-Capizzano, *Spectral analysis of the anti-reflective algebras and applications*, Linear Algebra Appl., 2/3 (2008), pp. 657–675.

[5] A. Beck, A. Ben-Tal, and M. Teboulle, *Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 425–445.

[6] A. Beck and A. Ben-Tal, *A global solution for the structured total least squares problem with block circulant matrices*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 238–255.

[7] A. Beck and A. Ben-Tal, *On the solution of the Tikhonov regularization of the total least squares problem*, SIAM J. Optim., 17 (2006), pp. 98–118.

[8] A. Beck and M. Teboulle, *A convex optimization approach for minimizing the ratio of indefinite quadratic functions over an ellipsoid*, Math. Program. Ser. A, to appear.

[9] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[10] D. F. Elliott and K. R. Rao, *Fast Transforms*, Academic Press, New York, 1982.

[11] R. D. Fierro, G. H. Golub, P. C. Hansen, and D. P. O'Leary, *Regularization by truncated total least squares*, SIAM J. Sci. Comput., 18 (1997), pp. 1223–1241.

[12] H. Fu and J. Barlow, *A regularized structured total least squares algorithm for high-resolution image reconstruction*, Linear Algebra Appl., 391 (2004), pp. 75–98.

[13] G. H. Golub, P. C. Hansen, and D. P. O'Leary, *Tikhonov regularization and total least squares*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 185–194.

[14] G. H. Golub and C. F. Van Loan, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.

[15] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, Fundam. Algorithms 3, SIAM, Philadelphia, PA, 2006.

[16] P. C. Hansen and D. P. O'Leary, *The use of the L-curve in the regularization of discrete ill-posed problems*, SIAM J. Sci. Comput., 14 (1993), pp. 1487–1503.

[17] P. C. Hansen, *Regularization tools, a matlab package for analysis of discrete regularization problems*, Numer. Algorithms, 6 (1994), pp. 1–35.

[18] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, PA, 1998.

[19] S. Van Huffel and J. Vandewalle, *Analysis and properties of the generalized total least squares problem $AX \approx B$ when some or all columns in A are subject to error*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 294–315.

[20] S. Van Huffel and J. Vandewalle, *The Total Least-Squares Problem: Computational Aspects and Analysis*, Frontiers Appl. Math. 9, SIAM, Philadelphia, PA, 1991.

[21] S. Van Huffel and H. Zha, *The restricted total least squares problem: Formulation, algorithm, and properties*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 292–309.
[22] P. Lemmerling and S. Van Huffel, *Analysis of the structured total least squares problem for Hankel/Toeplitz matrices*, Numer. Algorithms, 27 (2001), pp. 89–114.
[23] I. Markovsky, S. Van Huffel, and R. Pintelon, *Block-Toeplitz/Hankel structured total least squares*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1083–1099.
[24] I. Markovsky and S. Van Huffel, *On weighted structured total least squares*, in Large-Scale Scientific Computing, Lecture Notes in Comput. Sci. 3743, Springer-Verlag, Berlin, 2006, pp. 695–702.
[25] N. Mastronardi, P. Lemmerling, and S. Van Huffel, *Fast structured total least squares algorithm for solving the basic deconvolution problem*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 533–553.
[26] N. Mastronardi, P. Lemmerling, A. Kalsi, D. P. O'Leary, and S. Van Huffel, *Implementation of the regularized structured total least squares algorithms for blind image deblurring*, Linear Algebra Appl., 391 (2004), pp. 203–221.
[27] V. Z. Mesarovic, N. P. Galatsanos, and A. K. Katsaggelos, *Regularized constrained total least squares image restoration*, IEEE Trans. Image Process., 4 (1995), pp. 1096–1108.
[28] B. De Moor, *Total least squares for affinely structured matrices and the noisy realization problem*, IEEE Trans. Signal Process., 42 (1994), pp. 3104–3113.
[29] J. J. Moré, *Generalizations of the trust region subproblem*, Optim. Methods Softw., 2 (1993), pp. 189–209.
[30] M. K. Ng, R. H. Chan, and W.-C. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866.
[31] M. K. Ng, R. J. Plemmons, and F. Pimentel, *A new approach to constrained total least squares image restoration*, Linear Algebra Appl., 316 (2000), pp. 237–258.
[32] A. Pruessner and D. P. O'Leary, *Blind deconvolution using a regularized structured total least norm algorithm*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 1018–1037.
[33] R. A. Renaut and H. Guo, *Efficient algorithms for solution of regularized total least squares*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 457–476.
[34] J. B. Rosen, H. Park, and J. Glick, *Total least norm formulation and solution for structured problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 110–126.
[35] S. Serra-Capizzano, *A note on antireflective boundary conditions and fast deblurring models*, SIAM J. Sci. Comput., 25 (2003), pp. 1307–1325.
[36] D. Sima, S. Van Huffel, and G. H. Golub, *Regularized total least squares based on quadratic eigenvalue problem solvers*, BIT, 44 (2004), pp. 793–812.
[37] H. Zha, *The restricted singular value decomposition of matrix triplets*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 172–194.
[38] http://iew3.technion.ac.il/~becka/papers/rstls_package.zip

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 445–463

ON THE COMPUTATION OF NULL SPACES OF SPARSE RECTANGULAR MATRICES*

CRAIG GOTSMAN† AND SIVAN TOLEDO‡

Abstract. Computing the null space of a sparse matrix is an important part of some computations, such as embeddings and parametrization of meshes. We propose an efficient and reliable method to compute an orthonormal basis of the null space of a sparse square or rectangular matrix (usually with more rows than columns). The main computational component in our method is a sparse LU factorization with partial pivoting of the input matrix; this factorization is significantly cheaper than the QR factorization used in previous methods. The paper analyzes important theoretical aspects of the new method and demonstrates experimentally that it is efficient and reliable.

Key words. null space, inverse iteration, rectangular matrices, LU factorization, sparse matrices polynomial

AMS subject classifications. 15A18, 65F15, 65F30, 65F50

DOI. 10.1137/050638369

1. Introduction.

We propose a new method for computing an orthonormal basis for the null space of a rectangular m-by-n matrix A with $m \geq n$. The main computational component of the new method is a conventional LU factorization with partial pivoting of A. For many classes of sparse matrices, an appropriate preordering of the columns leads to sparse factors, usually significantly sparser than the QR factors and than the factors of any rank-revealing factorization [24, 26]. There are several recent high-quality sparse LU codes that can perform partial pivoting [2, 18, 19, 20, 31, 32]. Experts on sparse-matrix factorizations, including experts on sparse QR, believe that sparse LU factorizations with partial pivoting are intrinsically cheaper than sparse QR.[1] This belief is driven both by theoretical results and by computational experience. The theoretical results show that the nonzero pattern of the R factor in the QR factorization of a square matrix with no zeros on the diagonal contains the nonzero pattern of the L and U factors in the LU factorization with partial pivoting, so the L and U factors are at most as dense as R [23]. Therefore, the new method is particularly suitable for large sparse matrices with a small-dimensional null space. (Because we compute an orthonormal basis, in the computation of a high-dimensional null space, the cost of orthogonalizing the null vectors dominates.)

We also revisit a somewhat more expensive method, which is based on a QR factorization. This method is not new, but is not widely known either. Because of this, and because it can be used to compute not only the null space, but also additional singular triplets corresponding to small singular values, we mention it here briefly too.

*Received by the editors August 18, 2005; accepted for publication (in revised form) by J. H. Brandts August 13, 2007; published electronically May 2, 2008.
http://www.siam.org/journals/simax/30-2/63836.html

†Department of Computer Science, Technion Israel Institute of Technology, Haifa 32000, Israel (gotsman@cs.technion.ac.il).

‡School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel (stoledo@tau.ac.il). This author's research was partially supported by an IBM Faculty Partnership Award, by grant 848/04 from the Israel Science Foundation (founded by the Israel Academy of Sciences and Humanities), and by grant 2002261 from the United-States-Israel Binational Science Foundation.

[1]Private discussions with John Gilbert, Pontus Matstoms, and Esmond Ng in April and June 2005.

There are several applications to the computation of the null space and, more generally, to the computation of singular vectors or singular subspaces associated with small singular values. The first application that we describe involves computation with matrices derived from graphs. We start with a discussion of the square and symmetric case, and then describe extensions of these applications to rectangular matrices.

The spectrum of matrices related to the adjacency matrix of a graph is extremely interesting. Typically the lowest eigenvalues and eigenvectors are the most useful, as they characterize various properties of the graph. The most famous example is the second smallest eigenvalue of the graph Laplacian, which characterizes how strongly it is connected, hence its mixing rate [13]. The smallest eigenvalue is zero, corresponding to a fixed-valued eigenvector. The eigenvector corresponding to the second eigenvector (the so-called Fiedler vector) is also useful for ordering the vertices of the graph (using the components of this vector) for embedding [33] and partitioning [1] purposes. In general, the d eigenvectors corresponding to the d smallest nonzero eigenvalues may be used to form partitions and embed in $\mathbb{R}^d$. Singular vectors associated with the smallest singular values also play an important role in techniques for embedding graphs in $\mathbb{R}^d$. For example, the null space of the so-called Colin de Verdière matrices [16, 29, 37] are used in convex embeddings of closed manifold genus-0 graphs in $\mathbb{R}^d$, and the null space of stress matrices [17] is used in unique (up to rigid transformations) embeddings of a graph with given edge lengths. When the graphs arise from a three-dimensional (3D) mesh structure, as frequently happens in computer graphics applications, or from a k-nearest-neighbor graph, as in feature-learning applications, the graphs tend to be very sparse, and the sparsity should be exploited in the computation of the small singular subspace.

Another area where the null space of rectangular matrices arises is the parameterization of manifold 3D meshes of genus $g > 0$ [28, 30, 44]. In this application, a graph is considered a discrete version of a vector field on a surface, and a discrete version of the one-form is defined for it. Of particular interest are the so-called harmonic one-forms, which satisfy certain balance conditions. The search for harmonic one-forms on a given mesh graph results in the formulation of a set of linear equations for unknowns corresponding to the edges of a graph. Some of the equations are derived for the edges incident on vertices, and some are derived from the edges bounding faces. The size of the matrix is N-by-E, where E is the number of edges in the graph, and N is typically close to E, but is also influenced by the genus. These matrices are sparse and rectangular, and the nullity (the dimension of the nullspace) is typically $2g$.

Approximate null vectors are also used in at least two areas of numerical linear algebra. One area is condition-number estimation. The spectral condition number of a matrix A is the ratio of its extreme singular values $\sigma_{\max}/\sigma_{\min}$. Estimating the largest singular value is relatively easy; the hard part is estimating $\sigma_{\min}$. This is done almost invariably by trying to find a vector v with a unit norm such that $\|Av\|$ is small. When A is nearly singular, this problem is roughly equivalent to finding an approximate null vector. Note that in condition-number estimation, the estimate can be accurate enough even when v is not a good approximation of the singular vector associated with the smallest singular value. Therefore, the methods for finding such a v are not very similar to the methods that we describe here; condition-number estimators usually favor speed over accuracy; many of them are reliable in practice but may fail on some matrices. For further details, see [34, Chapter 15] and the references therein.

Approximate null vectors are also used in algorithms that compute rank-revealing factorizations and algorithms that solve rank-deficient least-squares problems without

a rank-revealing factorization [11, 12, 21, 39]. Here, too, a relatively inaccurate approximate null vector is good enough. On the other hand, some of the rank-revealing factorizations require approximate null vector of a sequence of nested upper-triangular matrices (the incrementally constructed factor of A). There are specialized incremental condition-number estimators for these applications [8, 9]; our approach is not efficient enough for such applications.

We focus on the $m \geq n$ matrices because the problems of computing a basis for the null space when $m > n$ and when $m < n$ are fundamentally different. When $m < n$ the nullity is at least $n-m$. When n is much larger than m, the nullity is high, and the space and time costs of computing an orthogonal basis for the null space are dominated by the usually dense basis vectors and by the cost of orthogonalizing them. Thus, when $m < n$ the standard approach is to compute a sparse but not necessarily orthogonal basis. This topic has been researched extensively and is outside the scope of this paper [7, 14, 15, 25]. When $m \geq n$ or when m is only slightly larger than n the nullity can be small and the orthogonal-basis algorithms that we discuss are appropriate.

The rest of the paper is organized as follows. The next section introduces inverse iteration. Section 3 introduces symmetric inverse iterations for nonnormal and for rectangular matrices, and in particular symmetric R iteration. Section 4 presents our main contribution, an LU-based symmetric inverse iteration. Section 5 describes the results of numerical experiments, and section 6 compares our work to previously published work. Section 7 presents some conclusions and open questions.

2. Inverse iteration. Given a square matrix A, inverse iteration repeatedly solves the equation $Ax^{(t)} = x^{(t-1)}/\|x^{(t-1)}\|$ for $x^{(t)}$. The starting vector $x^{(0)}$ can be random, although there are alternatives that often work a little better. When A is symmetric, the iteration converges to an eigenvector associated with the smallest eigenvalue of A (in absolute value). If the equation is solved using a backward stable factorization, such as QR or LU with partial pivoting, the iteration converges even if A is singular. In fact, if A is singular, then the iteration converges very quickly, in most cases in one or two iterations.

If the nullity of A is larger than one, we can start with an n-by-k matrix $Y^{(0)}$, and in each iteration we solve $AY^{(t)} = X^{(t-1)}$ for $Y^{(t)}$ and then orthonormalize the columns of $Y^{(t)}$ to produce $X^{(t)}$. If k is at least as large as the dimension of null(A), then the first $n-\text{rank}(A)$ columns of $X^{(t)}$ converge to an orthonormal basis of null(A). This technique is essentially the inverse version of simultaneous iteration or subspace iteration. The same idea applies to all the inverse iterations that we describe in the rest of the paper.

When A is square but not normal, the method often works, but it may also fail [35]. When A is not even square, standard inverse iteration does not apply at all.

One issue that is outside the scope of this paper, but should be mentioned, is overflows in inverse iteration. Suppose that we apply inverse iteration using a QR factorization, such that the exact R factor of A is

$$R = \begin{bmatrix} 0 & 1 & & & \\ & 0 & \ddots & & \\ & & \ddots & 1 & \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix},$$

and that due to rounding errors the computed factor is

$$\tilde{R} = \begin{bmatrix} \epsilon & 1 & & & \\ & \epsilon & \ddots & & \\ & & \ddots & 1 & \\ & & & \epsilon & 1 \\ & & & & \epsilon \end{bmatrix}.$$

Consider solving $Rx = \vec{1}$ (the right-hand side is the vector of all ones). As ϵ shrinks, the solution converges to $[\epsilon^{-n}\ \epsilon^{-(n-1)}\ \cdots\ \epsilon^{-1}]^T$. If there are no overflows, this is a good approximation of the exact null vector $[1\ 0\ \cdots\ 0]^T$, as expected. But obviously ϵ^{-n} is extremely likely to overflow. There are techniques to mitigate this danger [35], but they are difficult to apply in the case of subspace iteration. We shall see examples of this behavior in the numerical results below.

3. Symmetric iterations and symmetric inverse *R* iteration. When A is not normal or not even square, a variant of inverse iteration can still be applied reliably. This variant is not new, but not widely appreciated either. We call this variant [illegible] inverse iteration.

Symmetric power iteration repeatedly applies A^*A to a starting vector, and symmetric inverse iteration repeatedly solves equations of the form $A^*Ax^{(t)} = x^{(t-1)}/\|x^{(t-1)}\|$ for $x^{(t)}$. The Gram matrix A^*A is symmetric (Hermitian in the complex case), so inverse iteration works on it reliably. Symmetric iterations, both power and inverse, work without ever computing A^*A. The eigenvalues of A^*A are the squares of the singular values of A, so small singular values become a lot smaller: singular values near or below $\|A\|\sqrt{\epsilon_{\text{machine}}}$ become eigenvalues near or below $\|A\|\epsilon_{\text{machine}}$. If we compute A^*A explicitly, rounding errors usually make these small but nonzero singular values indistinguishable from the zero singular values of A, and inverse iteration will always produce linear combinations of the corresponding singular vectors. In other words, inverse iteration on an explicit A^*A may be unstable and may produce vectors that are far from null vectors of A.

But iterating on A^*A implicitly does not suffer from this instability. If A is square, we solve $A^*Ax^{(t)} = x^{(t-1)}/\|x^{(t-1)}\|$ by factoring A using any backward-stable factorization, say LU with partial pivoting, and solving

$$\begin{aligned} A^*w &= x^{(t-1)}/\|x^{(t-1)}\|, \\ Ax^{(t)} &= w\ . \end{aligned}$$

As far as we can tell, this idea is due to Stewart [42].

When A is not even square, we can still solve $A^*Ax^{(t)} = x^{(t-1)}/\|x^{(t-1)}\|$ if we compute the reduced QR factorization of A. In this factorization, Q is m-by-n (like A) with orthonormal columns, and R is n-by-n and upper triangular. Here, too, we need a backward-stable, but not rank-revealing, factorization. (So when A is sparse, we can use an arbitrary row and column preorderings to minimize fill and work.) Since $A = QR$, we have $A^*A = R^*Q^*QR = R^*R$. To solve $A^*Ax^{(t)} = R^*Rx^{(t)} = x^{(t-1)}/\|x^{(t-1)}\|$, we perform two triangular solves,

$$\begin{aligned} R^*w &= x^{(t-1)}/\|x^{(t-1)}\|, \\ Rx^{(t)} &= w\ . \end{aligned}$$

Björck [10, p. 109] credits Chan [12] with this technique, although Chan's paper is not explicit about how to carry out the inverse iteration. We refer to this technique as [illegible] R [illegible]. The simultaneous/subspace version of this technique can be used to compute multiple singular vectors associated with the smallest singular values of A.

Both the implicit normalization idea and the use of the R factor are not new, but neither are they widely appreciated. Virtually all the research on inverse iteration, surveyed by Ipsen [35], ignores normalization and focuses instead on less reliable and more complex methods to enhance inverse iteration for the nonnormal case. In that literature, there is essentially no discussion of rectangular matrices. So although normalization and the use of the R factor are mentioned in the literature, they are not widely known. For example, LAPACK [3] uses unsymmetric inverse iteration to compute eigenvalues of tridiagonal matrices [35].

4. *LU*-based symmetric inverse iterations. This section presents the main contribution of the paper, symmetric inverse iterations with the triangular factors computed by LU with partial pivoting. The algorithms start with a factorization

$$PA = LU = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} U , \tag{1}$$

where P is an m-by-m permutation matrix, L is an m-by-n upper trapezoidal matrix, and U is an n-by-n upper triangular matrix. Thanks to partial pivoting, L has 1's on the diagonal and the magnitude of all of its elements is bounded by 1. We partition L into pivot and nonpivot rows: L_1 is the square n-by-n triangular part of L, containing the pivot rows, and L_2 is the subdiagonal block containing the remaining $m-n$ rows.

In exact arithmetic, A, U, and L_1U all have exactly the same null space. Therefore, we can try to compute the null space of A by performing symmetric inverse iteration on U or on L_1U, both of which are square. The matrix U is upper triangular, and once we compute the factorization (1), we have a triangular factorization of L_1U. This allows us to perform symmetric inverse iteration with either L_1U or with U without any additional preprocessing. The following trivial lemma proves that A, U, and L_1U all have the same null space. Once we prove it, we analyze the effect of rounding errors on this process.

LEMMA 4.1. [illegible]

$$PA = LU = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} U$$

[illegible] LU [illegible] A [illegible] L_1 [illegible] $\operatorname{null}(A) = \operatorname{null}(L_1U) = \operatorname{null}(U)$

[illegible]. The row permutation P is irrelevant for null vectors, so without loss of generality we assume that $P = I$. Because L_1 is nonsingular, $\operatorname{null}(L_1U) = \operatorname{null}(U)$. Therefore, all we need to show is that $\operatorname{null}(A) = \operatorname{null}(L_1U)$. If $Ux = 0$, then we also have $L_2Ux = 0$, so $Ax = LUx = 0$. This shows that $\operatorname{null}(A) \subseteq \operatorname{null}(U) = \operatorname{null}(L_1U)$. On the other hand, if $Ax = LUx = 0$, then in particular $L_1Ux = 0$. This shows that $\operatorname{null}(A) \supseteq \operatorname{null}(L_1U) = \operatorname{null}(U)$, which concludes the proof. ∎

We now analyze the relationships between the approximate null spaces of A, U, and L_1U, still without taking into account rounding errors in the factorization of A. The relationships that we describe now motivate the structure and the numerics of our algorithm. The next theorem shows that

- all the small singular values of U correspond to small singular values of A (but A may have more small singular values than U);
- moreover, approximate null vectors of U are also approximate null vectors of A;
- all the small singular values of A correspond to small singular values of L_1U;
- approximate null vectors of A are approximate null vectors of L_1U.

THEOREM 4.2.

$$PA = LU = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} U$$

LU A L 1’ 1 x

$$\frac{3}{\sqrt{4^n + 6n - 1}}\sigma_k(U) \le \sigma_k(A) \le \sqrt{mn}\sigma_k(U), \tag{2}$$

$$\frac{3}{\sqrt{4^n + 6n - 1}}\|Ux\|_2 \le \|Ax\|_2 \le \sqrt{mn}\|Ux\|_2, \tag{3}$$

$$\sigma_k(L_1U) \le \sigma_k(A) \le \sqrt{1 + (m-n)n\frac{4^n + 6n - 1}{9}}\sigma_k(L_1U), \tag{4}$$

$$\|L_1Ux\|_2 \le \|Ax\|_2 \le \sqrt{1 + (m-n)n\frac{4^n + 6n - 1}{9}}\|L_1Ux\|_2. \tag{5}$$

(2) (3), $\sigma_k(U)$ $\sigma_k(A)$ $\sigma_k(U)$ m A U (2) (3) $\sigma_k(A)$ $2^n/3$ smaller $\sigma_k(U)$ A U U (4) (5) L_1U A A L_1U (4) (5) . We again assume without loss of generality that $P = I$.

The singular-value inequalities use the following singular-value bounds [6, Propositions 9.6.1 and 9.6.4], which hold for any $A = XY$ where X is m-by-ℓ and Y is ℓ-by-n:

$$\sigma_k(X)\sigma_\ell(Y) \le \sigma_k(A) \le \sigma_k(X)\sigma_{\max}(Y)\,. \tag{6}$$

We apply these bounds to two factorizations of A. When applied to $A^* = U^*L^*$, the inequalities gives

$$\sigma_k(U)\sigma_n(L) \le \sigma_k(A) \le \sigma_k(U)\sigma_{\max}(L)\,. \tag{7}$$

We also apply the bounds (6) to the conjugation of the factorization $A = M\,(L_1U)$, where

$$M = \begin{bmatrix} I \\ L_2L_1^{-1} \end{bmatrix}\,.$$

On $A^* = (L_1 U)^* M^*$, the singular-value bounds give

$$\sigma_k(L_1 U)\sigma_n(M) \le \sigma_k(A) \le \sigma_k(L_1 U)\sigma_{\max}(M) \; . \tag{8}$$

We now bound the smallest and largest singular values of L and M. Because the magnitude of all the elements of L are all bounded by 1, we have

$$\|L\|_2 \le \|L\|_F = \sqrt{\sum_{i,j} L_{i,j}^2} \le \sqrt{mn} \; .$$

The smallest singular value of L is bounded by that of L_1:

$$\begin{aligned}
\sigma_n(L) &= \min_{\|x\|_2=1} \|Lx\|_2 \\
&= \min_{\|x\|_2=1} \sqrt{\|L_1 x\|_2^2 + \|L_2 x\|_2^2} \\
&\ge \min_{\|x\|_2=1} \sqrt{\|L_1 x\|_2^2} \\
&= \sigma_n(L_1) \; .
\end{aligned}$$

Barlow and Zha [5, Lemmas 2.1 and 2.2] showed that

$$\sigma_n(L_1) \ge \left(\frac{4^n + 6n - 1}{9} \right)^{-1/2} \; ,$$

and this also bounds $\sigma_n(L)$. Combining these bounds with (7), we get

$$\sqrt{\frac{9}{4^n + 6n - 1}} \sigma_k(U) \le \sigma_k(A) \le \sqrt{mn}\sigma_k(U) \; .$$

We relate the singular values of A to those of $L_1^{-1} L_2$. The smallest singular value of M is at least 1, since

$$\begin{aligned}
\sigma_n(M) &= \min_{\|x\|_2=1} \|Mx\|_2 \\
&= \min_{\|x\|_2=1} \sqrt{\|Ix\|_2^2 + \|L_2 L_1^{-1} x\|_2^2} \\
&= \min_{\|x\|_2=1} \sqrt{1 + \|L_2 L_1^{-1} x\|_2^2} \\
&\ge 1 \; .
\end{aligned}$$

The largest singular value of M is bounded by

$$\begin{aligned}
\sigma_{\max}(M) = \|M\|_2 &= \max_{\|x\|_2=1} \sqrt{1 + \|L_2 L_1^{-1} x\|_2^2} \\
&\le \sqrt{1 + \|L_2 L_1^{-1}\|_2^2} \\
&\le \sqrt{1 + \|L_2\|_2^2 \|L_1^{-1}\|_2^2} \\
&\le \sqrt{1 + \|L_2\|_F^2 \sigma_n(L_1)^{-2}} \\
&\le \sqrt{1 + (m-n)n \frac{4^n + 6n - 1}{9}} \; .
\end{aligned}$$

Substituting the bounds on the singular values of M in (8), we get

$$\sigma_k(L_1U) \le \sigma_k(A) \le \sqrt{1 + (m-n)n\frac{4^n + 6n - 1}{9}}\sigma_k(L_1U) \ .$$

The bounds on approximate null vectors in the statement of the theorem are now easy to derive. One side follows from $\|Ax\|_2 \le \|L\|_2\|Ux\|_2$ (and similarly for $A = ML_1U$). The other side follows from

$$\sigma_n(L) = \min_y \frac{\|Ly\|_2}{\|y\|_2} \ ,$$

which imply that for $y = Ux$ we have $\sigma_n(L) \le \|Ly\|_2/\|y\|_2$, so $\sigma_n(L)\|Ux\|_2 \le \|Ly\|_2 = \|LUx\|_2 = \|Ax\|_2$. □

Rounding errors in the factorization do not loosen these bounds in a significant way, as long as the factorization is backward stable.

LEMMA 4.3. [illegible]

$$A + E = PLU = P\begin{bmatrix} L_1 \\ L_2 \end{bmatrix} U$$

[illegible] LU [illegible] m [illegible] n [illegible] A [illegible]

$$\|E\|_F \le \epsilon$$

[illegible] ϵ [illegible]

$$\begin{aligned} \sigma_k(A) &\le \sqrt{mn}\sigma_k(U) + \epsilon, \\ \|Ax\|_2 &\le \sqrt{mn}\|Ux\|_2 + \epsilon\|x\|_2, \\ \sigma_k(L_1U) &\le \sigma_k(A) + \epsilon, \\ \|L_1Ux\|_2 &\le \|Ax\|_2 + \epsilon\|x\|_2 \ . \end{aligned}$$

[illegible]. We again assume without loss of generality that $P = I$.

The standard perturbation theory for singular values ensures that [27, Theorem 8.6.4]

$$\sum_{k=1}^{n} (\sigma_k(A+E) - \sigma_k(A))^2 \le \|E\|_F^2 \ .$$

Therefore, $\sigma_k(A+E) - \sigma_k(A) \le \epsilon$, so using (2) we obtain

$$\sigma_k(A) \le \sigma_k(A+E) + \epsilon \le \sqrt{mn}\sigma_k(U) + \epsilon \ .$$

As for the approximate null vectors, we have

$$\begin{aligned} \|Ax\|_2 &= \|LUx - Ex\|_2 \\ &\le \|LUx\|_2 + \|Ex\|_2 \\ &\le \sqrt{mn}\|Ux\|_2 + \|E\|_2\|x\|_2 \\ &\le \sqrt{mn}\|Ux\|_2 + \|E\|_F\|x\|_2 \\ &\le \sqrt{mn}\|Ux\|_2 + \epsilon\|x\|_2 \ . \end{aligned}$$

Clearly, for this bound we could have used a bound on the 2-norm of E.

The proofs for the bounds on L_1U are similar. □

These results suggest the following algorithm.

1. Compute an LU factorization with partial pivoting $PA = LU = \left[\begin{smallmatrix} L_1 \\ L_2 \end{smallmatrix}\right] U$.
2. Perform subspace symmetric inverse iteration with U to find k approximate null vectors of U and to estimate the value of $\sigma_{k+1}(U)$. By Lemma 4.3, the approximate null vectors are also approximate null vectors of A, but there may be more.
3. Estimate $\sigma_n(L_1)$ using symmetric inverse iteration with L_1. If $\sigma_n(L_1)\sigma_{k+1}(U)$ is large, then

$$\sigma_{k+1}(A) \geq \sigma_n(L)\sigma_{k+1}(U) \geq \sigma_n(L_1)\sigma_{k+1}(U)$$

   is also large, so there is no need to search for additional approximate null vectors of A. If $\sigma_n(L_1)$ is too small, go to step 5; otherwise, continue.
4. L_1 is not ill conditioned: the vectors that we computed in step 2 should be a basis for the null space of A. Report the numerical rank of A and the basis for the null space, and return.
5. L_1 is ill conditioned enough to allow for approximate null vectors for A that are not approximate null vectors of U. Run subspace symmetric inverse iteration on $L'U$ to find a basis for its approximate null space.
6. If the iteration in step 5 produced approximate null vectors of $L'U$, then by Lemma 4.3 their number is an upper bound on the numerical rank deficiency of A (this number is possibly larger than the number of vectors found in step 2).
7. Determine which of the vectors produced in step 5 is also an approximate null vector of A and linearly independent of the vectors produced in step 2. Return these vectors, along with the vectors produced in step 2. Also report the upper bound computed in step 6.

The lower trapezoidal factor L that LU with partial pivoting produces is usually well conditioned, so we expect that the algorithm will usually perform steps 1–4 and stop there. If the algorithm does continue to steps 5–7, then the approximate null vectors that it returns may or may not constitute a basis for the null space of A. More specifically, if their number is smaller than the upper bound on the nullity, the vectors may span only a proper subspace of $\text{null}(A)$, or they constitute a basis but the upper bound is loose.

The details of steps 3 and 5 are as follows. Since step 3 is more complex (it needs to find an approximation to a nonzero singular value, in addition to finding a basis for the approximate null space), we explain how it works. We use a simple version of subspace iteration (see [43] for details) in which we orthogonalize the basis in every iteration. We do not use Schur–Raleigh–Ritz refinement or deflation. We apply the iteration implicitly to $U^{-1}U^{-*}$. To ensure that U has an inverse, we replace zero diagonal entries by the value $\epsilon_{\text{machine}}\|U\|_1$. This amounts to a small perturbation that does not change the null space of U in a significant way. Since we are looking for approximate null vectors, we perform very few iterations within each subspace iteration, usually 3. We perform this iteration repeatedly, each time with twice as many basis vectors, starting from two. We stop this repeated doubling when not all the vectors returned by the iteration are approximate null vectors of U. When this happens, we use the Raleigh quotient of the first nonnull vector that is returned as an approximation to $\sigma_{k+1}(U)$. Other symmetric/Hermitian subspace eigensolvers can also be used as the inner iteration of these steps.

5. Numerical experiments. In this section we provide a few illustrative examples to demonstrate the behavior of the LU- and QR-based algorithms, including pathological behaviors.

We carried out the experiments using MATLAB version 7.0 on a 3 GHz Pentium 4 computer with 1 GB of main memory running Linux. The LU factorization was performed using the function call `[L,U,P,Q]=lu(A,1.0)`, which calls UMFPACK version 4.3 (`P` and `Q` are row and column permutations). This syntax enforces partial pivoting and allows the sparse factorization code to reorder rows and columns for sparsity (under the partial pivoting constraint). We also ran a symmetric inverse R iteration, to compare the runtimes. We computed the R factor using the function call `R=qr(A(:,colamd(A)),0)`, which avoids the expensive computation of an explicit Q. The reordering of the columns tends to reduce fill and work, and is generally similar to the column ordering that UMFPACK uses.

The codes implement the algorithms from sections 3 and 4. They apply the iterations first to 1 vector, then 2, then 4, and so on, until the dimension of the computed null space stops growing. The codes always run 3 iterations of the appropriate strategy, starting from a matrix consisting of uniformly distributed random numbers between 0 and 1.[2]

5.1. Accuracy. We created random matrices with singular values $1, \ldots, 1, \sigma_2, 0$ for $\sigma_2 = 10^{-16}, 10^{-15}, \ldots, 10^{0}$. The matrices are all 200-by-100, and they were computed by generating random orthonormal singular vectors and multiplying the singular vectors and singular values appropriately. We generated 100 random matrices for each σ_2. For each matrix, we used our algorithm to compute its null vector. We also used MATLAB's singular value decomposition (`svd`) on both A and $A^T A$. On all of these matrices, our algorithm computed the null space of A by iterating on U; L was never ill conditioned. The results of the experiment, shown in Figure 1, show that our algorithm is less accurate than a full SVD computation, but not significantly so. In particular, the results show that the qualitative behavior of our algorithm is similar to that of a full SVD: the accuracy degrades smoothly as σ_2 approaches $\epsilon_{\text{machine}}\|A\|$.

We also ran experiments on matrices whose U factors have tiny diagonal values near the upper left corner. We did this by generating two independent columns, then a column that depends on the first two, and then another column that almost depends on the first two, but not exactly. We then completed the matrices with 96 additional linearly independent columns. This yielded matrices with norm around 1, one zero singular value, and one singular value near 10^{-8}. On the diagonal of U we have U_{33} close to $\epsilon_{\text{machine}}$ and U_{44} is small. We have also conducted experiments in which the dependent and almost-dependent columns were columns 4 and 3, to swap the small and numerically zero elements on the diagonal of U. The accuracy in these experiments was similar to the accuracy achieved in the previous experiments. From this experiment it appears that the position of small elements on the diagonal of U does not have a significant influence on the accuracy of the algorithm.

5.2. Large matrices. We conduced experiments on a few large sparse matrices from Davis's sparse matrix collection.[3] More precisely, we took matrices from this collection and modified them slightly to make them rectangular and singular. This experiment serves three purposes. First, it shows that the algorithm runs reasonably quickly even on large matrices. Second, it shows that our LU-based algorithm is

[2]The main code, `nulls.m`, is publicly available at http://www.tau.ac.il/~stoledo/research.html.
[3]http://www.cise.ufl.edu/research/sparse/matrices/.

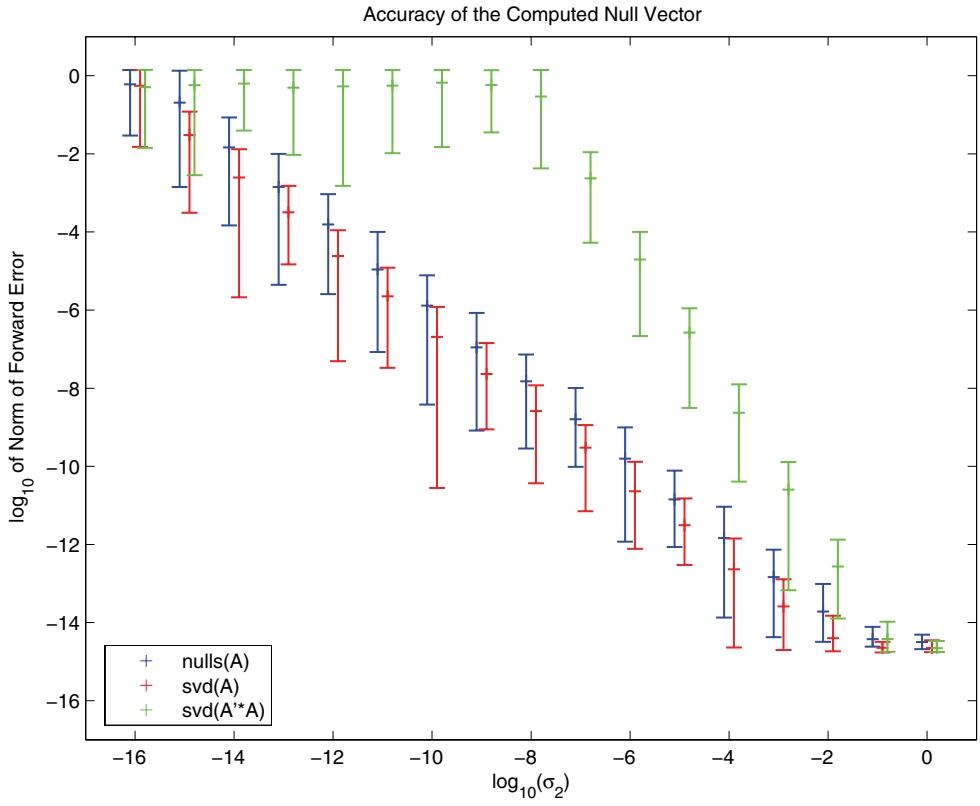


FIG. 1. *The results of the accuracy experiments. The bars show the range of accuracies for each* σ_2 *in* 100 *experiments; the mark along each range is the mean log accuracy. For each value of* σ_2*, the graph shows the accuracy of our algorithm (denoted* `nulls`*), of MATLAB's SVD implementation, and of the SVD applied to* A^TA*. The three bars for each* σ_2 *are slightly offset so that they don't overlap, but they all represent experiments with exactly the same* σ_2*.*

much faster than symmetric inverse R iteration. Third, it shows that the algorithm can fail; the failures are not specific to the LU-based algorithm but to inverse iteration in general. We believe that the failures are mostly due to scaling and overflow problems, similar to the ones discussed in section 2. In principle, these problems can be addressed by exploiting the capabilities of floating-point hardware better, but we have not implemented such measures.

We constructed the matrices as follows. All the matrices were initially square. From each matrix we dropped the first and last rows, and then duplicated rows 11 to 20 at the bottom of the matrix. This created $(n+8)$-by-n rectangular matrices with rank at most $n-2$.

The results of the experiments are summarized in Tables 1 and 2. Table 1 lists the matrices and the dimensions of the computed null spaces. Since the matrices were constructed to have null spaces of dimension at least 2, any dimension less than 2 indicates failure. Dimensions larger than 2 reflect matrices that were originally singular. One group of matrices, MULT_DCOP, caused difficulties to the LU-based algorithm, resulting in two failures. One of the failures led to overflows, but the other was silent. (The QR factorization of these matrices ran out of memory.) Two of these matrices, MULT_DCOP_02 and 03, have highly skewed row scaling, which may

TABLE 1

Our test matrices and the sizes of the computed null spaces. The columns denoted d display the dimensions of the computed null spaces, and the $\overset{NaN}{\infty}$ columns show whether any overflows or NaN's were detected during the iterations. The QR factorization ran out of memory on three matrices.

		LU		*QR*	
Base matrix	n	d	$\overset{\text{NaN}}{\infty}$	d	$\overset{\text{NaN}}{\infty}$
FPGA_TRANS_02	1220	2		2	
SHYY41	4720	4		0	Y
UTM5940	5940	2		2	
POISSON3DA	13514	2		2	
MULT_DCOP_01	25187	1		—	
MULT_DCOP_02	25187	2		—	
MULT_DCOP_03	25187	0	Y	—	
WANG4	26068	2		2	
ONETONE1	36057	2		2	
TWOTONE	120758	2		2	

TABLE 2

Runtimes and the size of the factors. The columns denoted T display the total running times, and the columns denoted T_f show the running time of the factorization alone. The columns η_L, η_U, and η_R show the number of nonzeros in the computed factors. The QR factorization ran out of memory on three matrices.

		LU				*QR*		
Base matrix	n	T	T_f	η_L	η_U	T	T_f	η_R
FPGA_TRANS_02	1220	0.09	0.03	5.7e3	6.1e3	0.14	0.05	2.8e4
SHYY41	4720	0.76	0.13	5.9e4	7.1e4	5.24	0.34	1.8e5
UTM5940	5940	1.87	0.50	3.8e5	4.9e5	5.04	3.27	8.8e5
POISSON3DA	13514	23.84	7.27	5.9e6	6.0e6	290.67	255.85	1.7e7
MULT_DCOP_01	25187	2.18	1.06	1.4e5	3.5e5	—	—	—
MULT_DCOP_02	25187	2.82	0.96	1.0e5	3.2e5	—	—	—
MULT_DCOP_03	25187	3.56	1.26	1.2e5	3.4e5	—	—	—
WANG4	26068	47.94	15.48	1.1e7	1.1e7	478.16	431.35	2.3e7
ONETONE1	36057	14.94	5.03	1.8e6	2.5e6	45.62	36.42	4.3e6
TWOTONE	120758	29.77	12.75	3.2e6	4.8e6	132.66	98.60	1.6e7

contribute to the difficulty: the ratio between the extreme ∞-norms of rows is 10^{12} for MULT_DCOP_03, and even larger for 02. Another matrix, SHYY41, which was originally singular, caused similar difficulties for the QR-based algorithm. This shows that this class of numerical difficulties is not associated with our new LU-based algorithm, but with inverse iteration in general.

Table 2 shows the performance of the two algorithms. On all the matrices, the LU-based algorithms ran in less than 30 seconds. On several large matrices it ran in less than 10 seconds. We argue that these are acceptable running times. The table also shows that in all the experiments, the QR-based algorithm was slower, in most cases substantially slower. This is probably due both to the fact that MATLAB 7 uses a state-of-the-art sparse LU factorization but a much older sparse QR, and to the intrinsic differences in the costs of sparse LU and QR factorizations. A comparison of the fill in the LU factors and the fill in the QR factor shows that the R factor is denser, but not significantly more than L and U combined.

5.3. Extreme examples. We now describe matrices that cause extreme behaviors in inverse-iteration algorithms. Experiments with these matrices constitute a partial coverage test of our implementation, because they exercise parts of the code that are rarely reached on real-world matrices.

We start with a particularly pathological matrix, suggested to us as an example by G. W. Stewart. This matrix has the form

$$A_S = \begin{bmatrix} 1 & & & & & \\ -1 & 1 & & & & \\ \vdots & & \ddots & & & \\ -1 & & & 1 & & \\ -1 & -1 & \cdots & -1 & 1 & \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix}.$$

The matrix A_s is $(n+1)$-by-n, has 1's on the diagonal, -1 below the diagonal in rows 1 through n, and 0.5 in all the entries in row $n+1$. The LU factorization with partial pivoting of A_S is $A_S = A_S I$, because A_S is already lower trapezoidal and its subdiagonal entries are bounded by 1 in absolute value. This matrix is well conditioned, so symmetric inverse iterations should not find any approximate null vectors. Indeed, if we perform symmetric inverse iteration with the upper triangular factor, we find no approximate null vectors. However, L_1, consisting of the first n rows of A_S, is very ill conditioned, since $L_1(1\ 2\ 4\ \ldots\ 2^n)^T = (1\ 1\ 1\ \ldots\ 1)^T$. This implies that the condition number of L' is exponential in n. When we run symmetric inverse iteration on either L_1 or L_1U, we find an approximation of the small singular vector of L_1. In this particular case, the large condition number of L_1 will cause our algorithm to iterate on $L_1 I$. This will return a single candidate vector and an upper bound of 1 on the rank deficiency. In this particular case, since the upper bound is 1, there is only one candidate vector x that is easy to rule out by observing that $\|A_S x\|$ is large. But this example shows that the exponential bound shown in Lemma 4.3 can be attained, and it shows that L_1 can be ill conditioned. The exponential bound implies that the dimension of the null space computed by iterating on L_1U is only an upper bound, and the ill conditioning of L_1 shows that the dimension of the null space computed by iterating on U is only a lower bound. Put together, this means that the method may fail to reliably estimate the rank deficiency (but it will report this failure explicitly, because it will detect the ill conditioning of L_1).

We also ran the algorithms on a block matrix of the form

$$\begin{bmatrix} A_S & 0 \\ 0 & A_R \end{bmatrix},$$

where A_S is the matrix describe above, and A_R is a random matrix with given singular values: all 1's except for four, which are three 0's and one 10^{-8}. On this matrix the QR-based algorithm correctly computes the rank deficiency, 3, and null space correctly, which is essentially the null space of A_R. The LU-based algorithm performs all the steps in the algorithm (that is, it does not stop at step 4 because it correctly detects that L' is ill conditioned). It finds three null vectors using inverse iteration on U, but since L_1U has four approximate null vectors, the algorithm returns the three null vectors but reports that the rank deficiency might be 4. In this particular case it is possible to determine the null space correctly, of course, but the example shows that the algorithm may need to resort to reporting a too-lax upper bound on the deficiency.

The next example shows that normalization may be necessary. The class of square matrices

$$A_I = \begin{bmatrix} 1 & \eta & & \\ & 1 & \ddots & \\ & & \ddots & \eta \\ & & & 1 \end{bmatrix},$$

where $\eta > 1$ is a parameter, was used by Ipsen [35] to show that without normalization, inverse iteration may fail. Their inverses are

$$A_I^{-1} = \begin{bmatrix} 1 & \eta & \eta^2 & \cdots & \eta^{n-1} \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \eta^2 \\ & & & \ddots & \eta \\ & & & & 1 \end{bmatrix}.$$

The norm of the matrices is $O(1+\eta)$ but the norm of the inverses is $O(1+\eta^{n-1})$. Therefore, the matrices are highly ill conditioned, so inverse iteration methods should find a vector x such that $A_I x$ has a small norm. However, without normalization, inverse iteration fails. With normalization, inverse iteration works. (Since A_I is upper triangular, iterating with A_I or with its R factor or with its U factor are exactly equivalent methods.)

5.4. Embeddings graphs on surfaces. We have also performed experiments on the following class of matrices. We describe the application where they arise, but we do not provide detailed results, since we detected no surprising or interesting behaviors on these matrices.

An instance of a nonnormal matrix whose null space is of interest is derived from a graph $G = (V, E, F)$ which has been embedded on a closed manifold surface of genus $g > 0$ in $\mathbb{R}^3$ (e.g., a torus). The graph G has $|V|$ vertices, $|E|$ edges, and $|F|$ faces. A value x_h may be attached to each half-edge h of G, such that $x_h = -x_{t(h)}$, where $t(h)$ is the (opposite) twin half-edge of h. Given an orientation for each edge, the vector x of the values corresponding to the half-edges in this orientation is known as a [illegible], or just one-form for short, of G [30]. A [illegible] one-form is one which satisfies some balance conditions, derived from each vertex and face of G. For a set of symmetric weights $w_h = w_{t(h)}$, each vertex v induces the [illegible] linear equation on x,

$$\sum_{h \in \delta v} w_h x_h = 0 ,$$

where δv is the set of half-edges emanating from v. Each face f induces the closedness linear equation,

$$\sum_{e \in \partial f} x_h = 0 ,$$

where ∂f is the set of half-edges bounding f.

In total, there are $|V| + |F|$ equations in $|E|$ unknowns, whose rank turns out to be $|V| + |F| - 2$. The Euler–Poincaré formula for manifold graphs asserts that $|V| + |F| - |E| = 2 - 2g$, so this rank is $|E| - 2g$. Thus, solving for a basis for the subspace of harmonic one-forms involves computation of the $2g$-dimensional nullspace of a nonnormal matrix of size $|E| + 2 - 2g$ by $|E|$.

By integrating harmonic one-forms, it is possible to parameterize manifold mesh data very efficiently. This has many applications in computer graphics and geometry processing [28, 30, 44].

6. Related work. Unsymmetric inverse iteration for square matrices is a well-researched area. The method was invented by Wielandt in 1944 and was studied by Wilkinson, who published his findings in several papers and books over a period of almost 30 years. For a comprehensive survey of these results, along with many newer results, see Ipsen's survey [35].

Symmetric inverse iteration for square matrices seems to have been first proposed by Stewart [42]. Symmetric inverse R iteration is due to Chan [12] (see also [10, p. 109]).

Symmetric inverse iteration with U or with L_1U is, to the best of our knowledge, new. It is remotely related to an idea by Saunders [40] to use U to precondition an iterative least-squares solver. There are additional least-squares preconditioners that are based on an LU factorization, but they all use the L'' block as well, so they are not really related to our proposed method (see [10, section 7.5.3] and the references cited there). Our algorithm is also related to the Peters–Wilkinson family of methods for solving least-squares problems using an LU factorization [38], in that both rely on the fact that L is usually well conditioned.

For square matrices, Schwetlick and Schnabel [41] proposed a bordering iteration as an alternative to inverse iteration. The advantage of their method is that the linear systems that their method solves in each iteration is nonsingular, so they can potentially be solved by an iterative linear solver, such as GMRES. However, the method is limited to square matrices that are numerically rank deficient by only one.

Friedman [22] proposed an algorithm to compute the null space of a symmetric positive definite matrix by shifting it so that the 0 eigenvalue, if there is one, is shifted to become the largest in magnitude, and then to apply power iteration. The motivation for the algorithm is related to the application that motivated our research, the application that we mentioned in section 5.4. Friedman's algorithm works as follows (we explain how to use it to compute the null space of A^*A). First, the algorithm uses power iteration on A^*A to compute an estimate $\tilde{\lambda}$ to the largest eigenvalue λ_n of A^*A. The estimate can be fairly inaccurate, but it should satisfy

$$\frac{2}{3}\lambda_n \le \tilde{\lambda} \le 2\lambda_n$$

with high probability. Now the algorithm performs power iteration on $B = \tilde{\lambda}I - A^*A$. The eigenvalues μ_i of B are $\mu_i = \tilde{\lambda} - \lambda_i$, where λ_i are the eigenvalues of A^*A. In particular, the zero eigenvalue of A^*A, if there is one, is shifted to $\tilde{\lambda}$, along with the invariant subspace. All the other μ_i's are smaller in magnitude than $\tilde{\lambda}$, by at least $\min(1/3, \lambda_{k+1})$, where λ_{k+1} is the smallest nonzero eigenvalue of A^*A. Therefore, power iteration on B converges to a null vector of A. The trouble with this approach is that the convergence is very slow if λ_{k+1} is small. This power iteration converges

linearly with a convergence rate that is proportional to

$$\frac{\lambda_n(A^*A) - \lambda_{k+1}(A^*A)}{\lambda_n(A^*A)} = 1 - \frac{\sigma_{k+1}^2(A)}{\sigma_n^2(A)}.$$

Convergence will be slow if $\sigma_{k+1}(A)$ is smaller than $\sigma_n(A)$, even if it is not dramatically smaller. The method does not converge at all to a null vector in floating point if $\sigma_{k+1}(A)/\sigma_n(A) \leq \sqrt{\epsilon_{\text{machine}}}$. The slow convergence or no convergence implies that the algorithm should probably be used only if (1) the only allowed use of A is multiplication by A and its conjugate, and (2) if it is known a priori that $\sigma_{k+1}(A)/\sigma_n(A)$ is relatively large, if at all.

The standard way to compute a basis for the null space of a rectangular matrix is using a rank-revealing factorization, such as a rank-revealing LU or QR factorization. For dense matrices, sophisticated rank-revealing factorizations are only slightly more expensive than backward-stable but non-rank-revealing ones. However, for sparse matrices, such factorizations can be significantly more expensive to compute than LU with partial pivoting or QR, because rank-revealing factorizations require column pivoting. In sparse QR and LU with partial pivoting, the column ordering is chosen so as to minimize fill and computation, so pivoting to reveal the rank typically leads to more fill and more work. Furthermore, sparse rank-revealing factorizations have not been implemented much, and those that have are not widely available. (The state of the art in this area is an algorithm by Pierce and Lewis [39], but the code is not publicly available; an earlier method proposed by Foster [21] uses similar techniques to detect dependent columns and to retriangularize R; see also [4].) In contrast, several recent and high-quality sparse LU with partial pivoting codes, which lie at the heart of our method, are publicly available [2, 18, 19, 20, 31, 32]. Some of these can exploit parallel computers and/or clusters. Even general-purpose interactive numerical engines, such as MATLAB, now contain excellent sparse LU codes (MATLAB 7 uses UMFPACK 4.3 [18]).

One method that might seem relevant but is not is inverse iteration on an augmented matrix

$$H = \begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix}.$$

The augmented matrix is symmetric (Hermitian), and its eigenvalues are the singular values of A with both signs and additional $m - n$ zero eigenvalues. The eigenvectors of H associated with nonzero eigenvalues are concatenations of left and right singular vectors of A. The difficulty is that the dimension of null(H) is larger by $m - n$ than the dimension of null(A). There is no straightforward way to compute a basis from null(A) from a basis for null(H). Also, if m is significantly larger than n, then H has a high-dimensional null space that is expensive to compute. This method is appropriate for computing bases for singular subspaces of A associated with a singular value $\sigma \gg 0$, by inverse iteration on $H - \sigma I$ (for σ close to zero, the null space of H causes inaccuracies in the computed singular vectors). MATLAB, for example, uses the augmentation idea in its sparse SVD routine `svds` (which fails when applied to the computation of the null space).

Finally, we mention that our ideas also apply to iterative Arnoldi/Lanczos-type algorithms. When these algorithms are used to find the smallest eigenpairs, they usually iterate on a representation of the inverse. This is the case, for example, in ARPACK [36], an Arnoldi-based package (ARPACK is the code that MATLAB's `eigs`

calls). Therefore, as in other forms of inverse iteration, the cost of these algorithms is likely to be dominated by the cost of factoring A. MATLAB's `eigs`, for example, calls exactly the same sparse LU factorization routine that our code calls. Also, if the inversion scheme is unsymmetric, these methods can suffer from the same problems that simple inverse iteration suffers from. This implies that applying Lanczos to the implicitly symmetric inverse is likely to be more reliable than applying Arnoldi to an unsymmetric inverse (this is possible when the application can use singular triplets rather than eigenpairs, which is the case when computing the null space).

We summarize the discussion of Arnoldi/Lanczos algorithms as follows. First, when applied to an unsymmetric inverse, they can be unreliable. We recommend that a symmetric inverse be used when using these algorithms to compute the null space. Second, our analysis in section 4 is also applicable when the null spaces of U, L', and possibly L_1U are computed using a symmetric Lanczos procedure rather than simultaneous inverse iteration.

7. Conclusions. We have shown how to utilize an LU factorization with partial pivoting of a nonnormal and possibly rectangular matrix to compute its null space. The algorithm is usually reliable and accurate. Furthermore, if the case of failure is ill conditioning in L_1, then it reports that it failed (rather than fail silently) and provides a reliable upper bound on the nullity, possibly along with a basis for a subspace of the null space.

Our new algorithm can also fail due to overflows or scaling problems, but this is a property of inverse iterations in general, not of this particular variant. These problems can be addressed by exploiting the capabilities of floating-point hardware, but our implementation does not take these measures. This makes the algorithm somewhat less reliable than rank-revealing factorizations, but it is also much cheaper.

Because our algorithm uses an LU factorization, it can be easily applied to large sparse matrices, using one of several available factorization codes. Relying on an LU rather than a QR factorization reduces the total cost, especially in the sparse case, where a QR factorization can be substantially more expensive to compute.

Our method can use a Lanczos iteration, rather than simple inverse iteration (to compute the null spaces of U, L_1, and possibly L_1U). The issue that our algorithm addresses is not the iteration itself, but the representation of the inverse, and the representations that we proposed are also applicable to Lanczos iterations.

Acknowledgments. Thanks to the two anonymous referees for constructive suggestions that helped improve the paper considerably. In particular, one of the referees suggested the elegant analysis given in Theorem 4.2. Our original analysis was done in a mixture of norms rather than just in the 2-norm, and was not as illuminating. The same referee also suggested using an estimate for $\sigma_{k+1}(U)$ in order to assess the degree of ill conditioning in L_1 that should cause the algorithm to iterate on L_1U. Our original criterion was not conservative enough when $\sigma_{k+1}(U)$ is small. Thanks to Pete Stewart and to Chen Greif for extensive comments on early drafts of this paper.

REFERENCES

[1] C. J. Alpert and S.-Z. Yao, *Spectral partitioning: The more eigenvectors, the better*, in DAC '95: Proceedings of the 32nd ACM/IEEE conference on Design automation, ACM Press, New York, 1995, pp. 195–200.

[2] P. R. Amestoy and C. Puglisi, *An unsymmetrized multifrontal LU factorization*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 553–569.

[3] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, 2nd ed., SIAM, Philadelphia, 1994; also available online from http://www.netlib.org.

[4] J. L. Barlow and U. B. Vemulapati, *Rank detection methods for sparse matrices*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1279–1297.

[5] J. L. Barlow and H. Zha, *Growth in Gaussian elimination, orthogonal matrices, and the 2-norm*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 807–815.

[6] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas with Applications to Linear Systems Theory*, Princeton University Press, Princeton, NJ, 2005.

[7] M. W. Berry, M. T. Heath, I. Kaneko, M. Lawo, R. J. Plemmons, and R. C. Ward, *An algorithm to compute a sparse basis of the null space*, Numer. Math., 47 (1985), pp. 483–504.

[8] C. H. Bischof, *Incremental condition estimation*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 312–322.

[9] C. H. Bischof, J. G. Lewis, and D. J. Pierce, *Incremental condition estimation for sparse matrices*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 644–659.

[10] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[11] T. F. Chan, *Deflated decomposition of solutions of nearly singular systems*, SIAM J. Numer. Anal., 21 (1984), pp. 738–754.

[12] T. F. Chan, *Rank revealing QR factorizations*, Linear Algebra Appl., 88/89 (1987), pp. 67–82.

[13] F. R. K. Chung, *Spectral Graph Theory*, CBMS Regional Conf. Ser. in Math. 92, AMS, Providence, RI, 1997.

[14] T. F. Coleman and A. Pothen, *The null space problem* I. *Complexity*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 527–537.

[15] T. F. Coleman and A. Pothen, *The null space problem* II. *Algorithms*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 544–563.

[16] Y. Colin de Verdière, *Sur un nouvel invariant des graphes et un critère de planarité*, J. Combin. Theory Ser. B, 50 (1990), pp. 11–21. English translation in Graph Structure Theory, N. Robertson and P. D. Seymour, eds., Contemp. Math. 147, AMS, Providence, RI, 1993, pp. 137–148.

[17] R. Connelly, *Rigidity and energy*, Invent. Math., 66 (1982), pp. 11–33.

[18] T. A. Davis, *A column pre-ordering strategy for the unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 165–195.

[19] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 720–755.

[20] J. W. Demmel, J. R. Gilbert, and X. S. Li, *An asynchronous parallel supernodal algorithm for sparse Gaussian elimination*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 915–952.

[21] L. V. Foster, *Rank and null space calculations using matrix decompositions without column interchanges*, Linear Algebra Appl., 74 (1986), pp. 47–71.

[22] J. Friedman, *Computing Betti numbers via combinatorial Laplacians*, Algorithmica, 21 (1998), pp. 331–346.

[23] A. George and E. Ng, *An implementation of Gaussian elimination with partial pivoting for sparse systems*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 390–409.

[24] A. George and E. Ng, *On the complexity of sparse QR and LU factorization of finite-element matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 849–861.

[25] J. R. Gilbert and M. T. Heath, *Computing a sparse basis for the null space*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 446–459.

[26] J. R. Gilbert and E. Ng, *Predicting structure in nonsymmetric sparse matrix factorizations*, in Graph Theory and Sparse Matrix Computation, A. George, J. R. Gilbert, and J. W. H. Liu, eds., Springer-Verlag, New York, 1993, pp. 107–139.

[27] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[28] S. J. Gortler, C. Gotsman, and D. Thurston, *Discrete one-forms on meshes and applications to 3D mesh parameterization*, Comput. Aided Geom. Design, 33 (2006), pp. 83–112.

[29] C. Gotsman, X. Gu, and A. Sheffer, *Fundamentals of spherical parameterization for 3D meshes*, ACM Trans. Graphics, 22 (2003), pp. 358–363.

[30] X. Gu and S.-T. Yau, *Computing conformal structures of surfaces*, Commun. Inf. Syst., 2 (2002), pp. 121–146.

[31] A. Gupta, *Improved symbolic and numerical factorization algorithms for unsymmetric sparse matrices*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 529–552.

[32] A. Gupta, *Recent advances in direct methods for solving unsymmetric sparse systems of linear equations*, ACM Trans. Math. Software, 28 (2002), pp. 301–324.

[33] K. M. Hall, *An r-dimensional quadratic placement algorithm*, Management Sci., 17 (1970), pp. 219–229.
[34] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[35] I. C. F. Ipsen, *Computing an eigenvector with inverse iteration*, SIAM Rev., 39 (1997), pp. 254–291.
[36] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.
[37] L. Lovász and A. Schrijver, *On the null space of a Colin de Verdière matrix*, Ann. Inst. Fourier (Grenoble), 49 (1999), pp. 1017–1026.
[38] G. Peters and J. H. Wilkinson, *The least-squares problem and pseudo-inverses*, Comput. J., 13 (1970), pp. 309–316.
[39] D. J. Pierce and J. G. Lewis, *Sparse multifrontal rank revealing QR factorization*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 159–180.
[40] M. A. Saunders, *Sparse least squares by conjugate gradients: A comparison of preconditioning methods*, in Proceedings of Computer Science and Statistics: 12th Annual Symposium on the Interface, J. F. Gentleman, ed., University of Waterloo, Waterloo, ON, Canada, 1979, pp. 15–20.
[41] H. Schwetlick and U. Schnabel, *Iterative computation of the smallest singular value and the corresponding singular vectors of a matrix*, Linear Algebra Appl., 371 (2003), pp. 1–30.
[42] G. Stewart, *Rank degeneracy*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 403–413.
[43] G. W. Stewart, *Matrix Algorithms, Volume* II: *Eigensystems*, SIAM, Philadelphia, 2001.
[44] G. Tewari, C. Gotsman, and S. Gortler, *Meshing genus-1 point clouds using discrete one-forms*, Computers and Graphics, 30 (2006), pp. 917–926.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 464–490

A QR-BASED SOLVER FOR RANK STRUCTURED MATRICES*

STEVEN DELVAUX† AND MARC VAN BAREL†

Abstract. In this paper we show how to compute the QR-factorization of a rank structured matrix in an efficient way by means of the Givens-weight representation. We also show how the QR-factorization can be used as a preprocessing step for the solution of linear systems. Provided the representation is chosen in an appropriate manner, the complexity of the QR-factorization is $O((ar^2+brs+cs^2)n)$ operations, where n is the matrix size, r is some measure for the average rank of the rank structure, s is some measure for the bandwidth of the unstructured matrix part around the main diagonal, and $a, b, c \in \mathbb{R}$ are certain weighting parameters. The complexity of the solution of the linear system with given QR-factorization is then only $O((dr+es)n)$ operations for suitable $d, e \in \mathbb{R}$. The performance of this scheme will be demonstrated by some numerical experiments.

Key words. rank structured matrix, Givens-weight representation, QR-factorization, structure inheritance, linear system solution

AMS subject classifications. 65F05, 65F25, 15A03

DOI. 10.1137/060654979

1. Introduction.

In this paper we describe how for a rank structured matrix with available Givens-weight representation, one can efficiently compute its QR-factorization and subsequently use this factorization for the solution of linear systems.

A matrix will be called [illegible] if the ranks of certain submatrices starting from its bottom left corner, as well as the ranks of certain submatrices starting from its top right corner, are small compared to the matrix size.

Although one can devise several ways to obtain a compact representation for a rank structured matrix, there seem to be two classes of representations frequently used in the literature, which we call *uv* [illegible] and [illegible], following the terminology of [3]. Let us give a brief survey.

The class of *uv*-representations was historically the first; see, e.g., [10]. We mention that this type of representation is possible only under certain conditions.

A different and more flexible class of block quasi-separable representations was introduced in the book of Dewilde and van der Veen [6]. Many algorithms for these representations have since been developed in the literature. After their initial appearance in [6], they were then used by Eidelman and Gohberg, who also introduced the name (block) "quasi-separable representation"; see, e.g., [8]. More recently, these

*Received by the editors March 24, 2006; accepted for publication (in revised form) by N. Mastronardi October 26, 2007; published electronically May 2, 2008. This research was partially supported by the Research Council K.U. Leuven, project OT/05/40 (Large Rank Structured Matrix Computations), Center of Excellence: Optimization in Engineering, by the Fund for Scientific Research–Flanders (Belgium), G.0455.0 (RHPH: Riemann–Hilbert Problems, Random Matrices and Padé–Hermite Approximation), G.0423.05 (RAM: Rational Modelling: Optimal Conditioning and Stable Algorithms), and by the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, project IUAP V-22 (Dynamical Systems and Control: Computation, Identification & Modelling). The scientific responsibility rests with the authors.

http://www.siam.org/journals/simax/30-2/65497.html

†Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), Belgium (Steven.Delvaux@cs.kuleuven.ac.be, Marc.VanBarel@cs.kuleuven.ac.be).

matrices appeared under the name of "sequentially semiseparable representations" in the work of Chandrasekaran et al.; see, e.g., [2].

In the paper [3], we introduced yet another type of representation for rank structured matrices, which we called the . This representation is a generalization of the so-called Givens-vector representation introduced in [12]. We showed that our representation is theoretically equivalent with the block quasi-separable representations in or described in the book [6]. Nevertheless, we note that we are not aware of any systematic treatment of the use of the latter representations in the literature. We also showed how the unitary-weight representation could be further fine-tuned to what we called the .

In the present paper, we consider the problem of solving linear systems with rank structured coefficient matrix. Many methods for doing this have already been described in the literature. At the risk of making a too crude distinction, these methods can be divided in at least three categories.

A first type of solvers is based on the Sherman–Morrison formula and its generalizations; see, e.g., [7, section 7] and [9, section 4] for the case of uv-representable matrices with low rank blocks lying just below the main diagonal.

A second type of solvers is based on LU-factorizations and Gaussian operations without pivoting, a fact which is often revealed by the condition that the matrix must be strongly nonsingular. See, e.g., [7, section 5] for the case of block quasi-separable matrices with low rank blocks lying just below the main diagonal, and see [9, section 5] for an algorithm in the uv-representable case.

Finally, a third type of solver is based on QR- or URV-factorizations. Such algorithms can be devised in a stable way and are therefore numerically superior to the other classes of solvers. The first algorithm of this kind was the URV-decomposition solver for block quasi-separable matrices, reported in a more general operator-theoretical context in the book [6, Chapter 7]. More efficient versions of this algorithm for the case of finite matrices were then obtained in [8] using the QR-decomposition (see also [11]), and more recently in [2] using a URV-decomposition in the case where the low rank blocks are situated just below the main diagonal. The latter algorithm is a generalization of earlier work on uv-representations originating from [1].

In the present paper, we follow the solution strategy of [8] by developing a linear system solver that is based on a preliminary QR-factorization. The algorithm will be expressed in terms of the Givens-weight representation. The computation of the QR-factorization will require about $O((ar^2 + brs + cs^2)n)$ operations, where n is the matrix size, r is some measure for the average rank of the rank structure, s is some measure for the bandwidth of the unstructured matrix part around the main diagonal, and $a, b, c \in \mathbb{R}$ are certain weighting parameters. This QR-factorization can then be used for the efficient solution of a linear system, the latter requiring only $O((dr+es)n)$ operations for suitable $d, e \in \mathbb{R}$.

The algorithm in this paper will be able to capture rank structure, irrespective of the position of the low rank blocks with respect to (w.r.t.) each other and to the main diagonal. The only restriction is that the low rank blocks in the block lower triangular part may not overlap with those in the block upper triangular part.

The remainder of this paper is organized as follows. In section 2 we recall the basic ideas of the Givens-weight representation from [3]. Section 3 considers the QR-factorization of a rank structured matrix. This section contains both a theoretical part concerning structure inheritance by the Q- and R-factors of the QR-factorization, as well as a practical part concerning the algorithmic exploitation of these inheritance

results. Section 4 deals with the linear system solver. Finally, section 5 reports on the results of some numerical experiments.

2. Givens-weight representation. In this section we review the basic ideas of the Givens-weight representation from [3].

First we define the class of rank structured matrices.

DEFINITION 1 (see, e.g., [5]). $\mathcal{R}$ $\mathbb{C}^{m\times n}$ $\mathcal{R} = \{\mathcal{B}_k\}_k$ $\mathcal{B}_k$ 3.

$$\mathcal{B}_k = (i_k, j_k, r_k),$$

i_k j_k r_k $A \in \mathbb{C}^{m\times n}$ $\mathcal{R}$ k

$$\text{Rank } A(i_k : m, 1 : j_k) \leq r_k.$$

$A(i_k : m, 1 : j_k)$ A $i_k, \ldots, m$ $1, \ldots, j_k$

The above definition uses the word to distinguish from the more general rank structures which were handled in [5]. Since these more general structures do not occur in the present paper, we will simplify notation by just dropping the word everywhere from the notation.

Note that by definition all structure blocks can be identified as contiguous blocks situated in the bottom left corner of the matrix. A pictorial illustration of a pure rank structure with two structure blocks is shown in Figure 2.1.

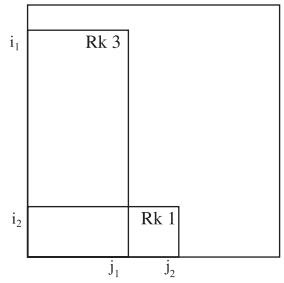


FIG. 2.1. *Example of a rank structure with two structure blocks $\mathcal{B}_1$ and $\mathcal{B}_2$. The notation "*Rk r_k*" denotes that the structure block is of rank at most r_k, $k = 1, 2$.*

In practice, it often happens that also the block triangular part is rank structured, i.e., that also the matrix A^T satisfies rank structure in the sense of Definition 1. By abuse of notation, we will indiscriminately use the term also in this case.

We will assume in what follows that we are working with a rank structure $\mathcal{R}$ for which there are no structure blocks that are "contained" in each other, i.e., for which the structure blocks $\mathcal{B}_k$ can be ordered such that both their row and column indices i_k and j_k increase in a strictly monotonic way. (Actually, these nested structure blocks are not completely useless, in the sense that they lead to an additional sparsity pattern in the Givens-weight representation, but we will not be concerned about this here.)

Now we will try to indicate the underlying ideas of the unitary-weight representation, following [3]. To this end we will take the structure in Figure 2.2 as a didactical

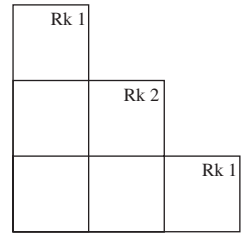


FIG. 2.2. *Example of a rank structure with three structure blocks, $\mathcal{B}_1, \mathcal{B}_2$, and $\mathcal{B}_3$. We will use this example to explain the mechanism of the unitary-weight and Givens-weight representations during the following paragraphs. From now on the surrounding matrix box, as in Figure* 2.1*, will not be shown anymore.*

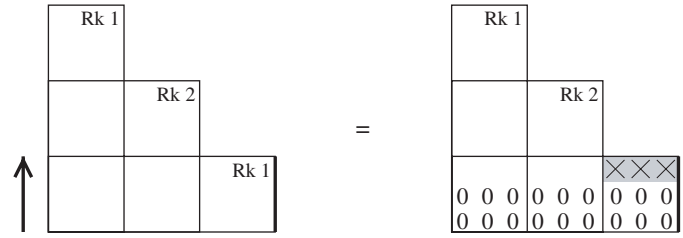


FIG. 2.3. *We apply a unitary transformation to transform the bottom two rows of the structure into zeros. This transformation acts only on the columns on the left of the vertical line, which is indicated in boldface in the left part of the figure; this line borders the* action radius *of the unitary transformation. Having performed this unitary transformation, the elements indicated on a gray background in the right part of the figure are stored; they are called* weights.

example. First, it may be noted that this figure does not show the surrounding matrix box anymore: This reflects the fact that only the area spanned by the structure blocks will be relevant for the representation, and that the "outside world" will be inaccessible.

In what follows, we will often work with [illegible]. These are defined as unitary matrices having a block diagonal form $U = I_a \oplus Q \oplus I_b$, where I_a, I_b denote identity matrices of suitable sizes a, b. When such a unitary operation U acts on the rows of a given matrix, we will represent it in a pictorial way by a vertical line segment, placed on the position of the rows on which it acts. (Sometimes we will actually denote it as a vertical [illegible], instead of a line segment, as an auxiliary means for visualizing the algorithm flow; see further.)

The unitary-weight representation is obtained by reducing the structure blocks into blocks of zeros by the use of unitary row transformations. First we apply an (elementary) unitary transformation to transform the bottom Rk 1 block into a block of zeros, with one row less; see Figure 2.3.

Note that this unitary transformation acts only on the columns on the left of the vertical line, which is indicated in boldface in the figure. We say that this line borders the [illegible] of the unitary transformation. Thus the action radius of the current unitary transformation is equal to 9.

Having applied this operation, note that in columns 7, 8, and 9 we have already reached the "top" of the structure. Therefore, this is now the right moment to consider the top elements of these columns, and to store them. These elements will be called [illegible], and they are visualized on a gray background in the right part of Figure 2.3.

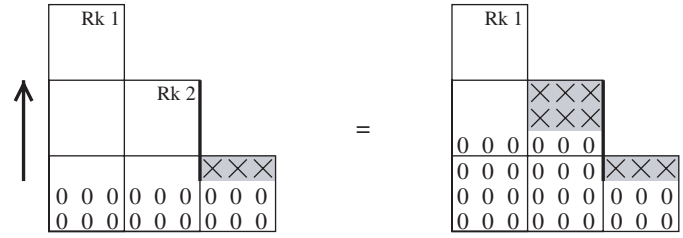


FIG. 2.4. *We apply the next unitary transformation, and store the new block of weights.*

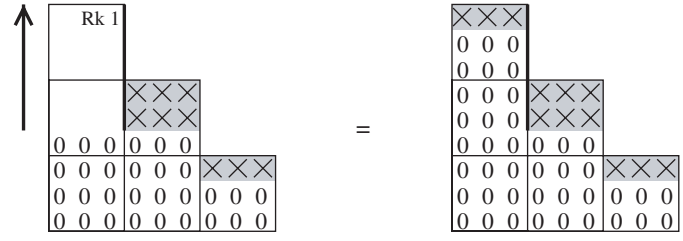


FIG. 2.5. *We apply the final unitary transformation, and store the new block of weights.*

From now on we consider columns 7, 8, and 9 as finished, and we restrict our perspective to the previous columns. We can then apply a unitary transformation to transform the middle Rk 2 block into a block of zeros, with two rows less; see Figure 2.4.

Note that again this unitary operation acts only on the columns on the left of the vertical line indicated in boldface in the figure. Thus the action radius of the current unitary transformation is equal to 6.

Having applied this operation, note that also in columns 4, 5, and 6 we have reached the top of the structure. Therefore, this is now the right moment to consider the top elements of these columns, and to store them. This yields us a second block of weights, which is again visualized on a gray background in Figure 2.4.

From now on we drop columns 4, 5, and 6 from our perspective. We can then apply a unitary transformation to transform the top Rk 1 block into a block of zeros, with one row less; see Figure 2.5. We conclude by storing the final block of weights.

The weights can now be collected into a single matrix, which we call the [illegible]. Together with the computed unitary transformations, this matrix yields us the complete [illegible] of the given matrix; see Figure 2.6.

Of course, to be a useful representation, the unitary-weight representation should allow the possibility of restoring the original matrix from which we started. This can be done by reversing the above steps. This reversal process is called [illegible] the unitary-weight representation and is described in [3].

Now we can come to the general definition of unitary-weight representations.

DEFINITION 2 (index sets). [illegible] $\mathcal{R} = \{\mathcal{B}_k\}_{k=1}^{K}$ [illegible] $i_1 < \cdots < i_K$ [illegible] $j_1 < \cdots < j_K$ [illegible] $I_k = \{i_k, \ldots, i_{k+1} - 1\}$, $I_{k,\text{top}} = \{i_k, \ldots, i_k + r_k - 1\}$, $J_k = \{j_{k-1}+1, \ldots, j_k\}$, $k = 1, \ldots, K$ [illegible] $i_{K+1} := N + 1$ [illegible] $j_0 := 0$ [illegible] $r_{K+1} := 0$

DEFINITION 3 (unitary-weight representation). [illegible] $A \in \mathbb{C}^{m \times n}$ [illegible] $\mathcal{R} = \{\mathcal{B}_k\}_{k=1}^{K}$ [illegible]

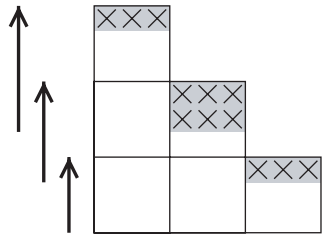
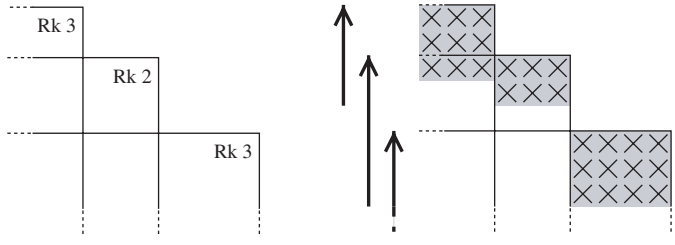


FIG. 2.6. *Schematic picture of the unitary-weight representation for the rank structure in Figure* 2.2.



(a) (b)

FIG. 2.7. *For the rank structure in the left picture, the right figure shows a schematic picture of the unitary-weight representation.*

$i_1 < \cdots < i_K$ $j_1 < \cdots < j_K$ unitary-weight representation A $\mathcal{R}$ $(\{U_k\}_{k=1}^K, W)$. U_k $I_k \cup I_{k+1,\text{top}}$ $\bigcup_{l=1}^k J_l$ $I_{k,\text{top}}$ U_k $k = K, K-1, \ldots, 1$ $W \in \mathbb{C}^{m \times n}$ weight matrix $I_{k,\text{top}}$ J_k U_k 2.7

We can now specify from unitary-weight to Givens-weight representations. In what follows, we will use the term to denote an elementary unitary operation which differs from the identity matrix only in two subsequent rows and columns i and $i+1$. This transformation will sometimes be denoted as $G_{i,i+1}$, and the index i will be called the of the Givens transformation. Similarly to our notation for elementary unitary operations, we will graphically denote the Givens transformation $G_{i,i+1}$ by means of a vertical line segment, with the height at which this line segment is standing in the figure determined by the row index i (see further).

Rather than individual Givens transformations, it will be useful to work with : these are defined as products of the form $G_{i+k,i+k+1} \ldots G_{i,i+1}$ for some $k \geq 0$. Graphically, this can be considered as a collection of Givens transformations where each Givens transformation is situated precisely one position below the previous one; see Figure 2.8.

The number of Givens transformations of which a Givens arrow consists will be called the of the Givens arrow. Moreover, we define the and the of the Givens arrow to be, respectively, the largest and the smallest row index of the Givens transformations of which the Givens arrow consists. These notions have an obvious graphical interpretation.

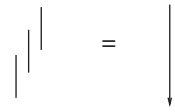


FIG. 2.8. *A Givens arrow $G_{i+2,i+3}G_{i+1,i+2}G_{i,i+1}$ consisting of* 3 *Givens transformations. Concerning this figure, we remind the reader that we consider each Givens transformation as "acting" on the rows of an (invisible) matrix standing on the right of it, and hence that the Givens transformations in the figure should be evaluated from right to left, hereby explaining the downward direction of the Givens arrow.*

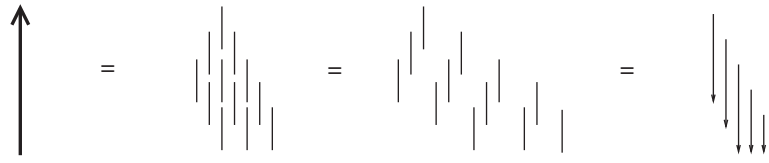


FIG. 2.9. *Suppose that the current structure block is* Rk 3*, and that the corresponding unitary transformation U_k spans over* 6 *rows. Then we assume for this unitary transformation a decomposition into a product of Givens arrows of width at most* 3.

DEFINITION 4 (Givens-weight representation [3]). $A \in \mathbb{C}^{m\times n}$ $\mathcal{R} = \{\mathcal{B}_k\}$ $i_1 < \cdots < i_K$ $j_1 < \cdots < j_K$ Givens-weight representation A $\mathcal{R}$ U_k

- r_k
- U_k

2.9

Let us comment on Figure 2.9. The left part of the figure denotes a unitary component U_k of the Givens-weight representation. The middle and rightmost part of the figure show then the required decomposition of this unitary component U_k into a product of Givens transformations. In particular, the equivalence between the two rightmost pictures in Figure 2.9 follows by repeatedly inserting Figure 2.8. On the other hand, the equivalence between the two pictures in Figure 2.9 is purely aesthetic: These are two different ways for visualizing the same product of Givens transformations.

We should still explain why the assumption is made that each Givens arrow in the decomposition of U_k has width at most r_k. To this end, recall that the unitary transformation U_k serves to create zeros in a certain $\mathrm{Rk}(r_k)$ submatrix, except for its top r_k rows. This effect can always be realized by a succession of Givens arrows as prescribed; see [3, section 3] for more details.

Note that by decomposing each unitary transformation U_k as specified in Definition 4, we formally obtain a decomposition into a product of Givens transformations, in the sense that the beginning and ending Givens transformations of two subsequent unitary transformations U_k may overlap. In relation to this, it was shown in [3] that one can always find a representation for the rank structured part containing only $O(rn)$ parameters, where n is the matrix size and r is some measure

for the average rank of the rank structure, and this applies to [illegible] distribution of structure blocks. However, since the description of the algorithms in this paper will remain exactly the same for all these cases, we will restrict ourselves to a full Givens distribution as in Figure 2.9, for reasons of clarity. (The more efficient Givens-weight representations in [3] follow then as a special case by taking some of these Givens transformations equal to the identity matrix.)

Finally, we note that the above results for obtaining a Givens-weight representation can be trivially extended to obtain a Givens-weight representation for the structured [illegible] triangular part of the matrix A. To this end, it suffices to apply the above results with A replaced by A^T. In this way, one will obtain a representation based on unitary [illegible] operations, rather than unitary row operations.

Suppose then that we have Givens-weight representations for the structured lower and upper triangular parts of the matrix A, respectively. We can then "glue" the weight matrices of these two representations together. This glued matrix will again be called the [illegible] W. An example is shown in Figure 3.6; note that the structured lower and upper triangular part are represented here by means of a row-based and a column-based Givens-weight representation, respectively. Note also that this figure shows a few real-size elements around the main diagonal, corresponding to the unstructured matrix part. These elements are distinguished from the actual weights by putting them on a white instead of a gray background in Figure 3.6.

3. QR-factorization. In this section we describe an algorithm that performs the QR-factorization of a rank structured matrix, assuming that there is available a Givens-weight representation for this matrix. The output of the algorithm consists of the Q- and R-factors of the QR-factorization, where the Q-factor is decomposed as a product of elementary unitary transformations, and where the R-factor has the form of a Givens-weight representation.

Before describing the algorithm, we will start with some theory concerning the structure which we may hope to exploit.

3.1. Some theory. In this subsection we recall and provide some theoretical results concerning the structure inheritance by the Q- and R-factors of a QR-factorization $A = QR$, where A is assumed to be a rank structured matrix.

The inheritance of structure by the Q-factor in terms of Givens transformations was handled in [4] (see also [6, 8] for related results). We considered there the matrix[1] Q^H as a product of Givens transformations "acting" on the matrix A, hereby transforming it into an upper triangular matrix $R = Q^H A$. This process proceeds in two phases.

For the first phase, we recall from section 2 that for a rank structured matrix A, a sequence of elementary unitary operations can be applied to transform the given Rk r_k structure blocks of A into blocks of zeros, except for their top r_k rows. This process proceeds from the bottom to the top of the matrix. We will call this the [illegible]

For the second phase, we note that the resulting matrix A at the end of the preparative phase will be almost zero in its lower triangular part, except for a few nonzero elements around the main diagonal. These remaining nonzero elements can then be annihilated by a sequence of upward-pointing Givens sequences, proceeding from the top to the bottom of the matrix. We will call this the [illegible]

[1] Throughout this paper, we use the notation Q^H to denote the Hermitian transpose of Q, i.e., the complex conjugate transpose.

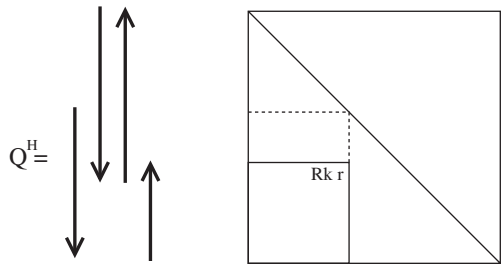


FIG. 3.1. *For the structure block $\mathcal{B}$ shown on the right, the left picture shows the corresponding sparsity pattern of the matrix Q^H w.r.t. this structure block, transforming it into an upper triangular matrix $Q^H A = R$. The picture shows a decomposition into preparative (upward) and residual (downward) unitary transformations.*

We note that the preparative and residual phase correspond to the so-called [illegible] and [illegible] in [8], respectively.

We now want to investigate the inheritance of structure by the upper triangular matrix $R = Q^H A$ obtained at the end of the residual phase. It turns out that we have to exclude some pathological cases.

DEFINITION 5 (structure implying rank deficiency). [illegible] $\mathcal{B} = (i, j, r)$ [illegible] $n_{\mathcal{B}} = \max\{0, j - i + 1\}$ [illegible] $\mathcal{B}$ [illegible] imply rank deficiency [illegible] $n_{\mathcal{B}} > r$ [illegible] $\mathcal{R} = \{\mathcal{B}_k\}_k$ [illegible]

It can be shown that a structure block implying rank deficiency is equivalent with the structure block [illegible], causing a linear dependency between the columns of the underlying matrix. In particular, in the case of square matrices the definition reduces to that of [illegible] as defined in [4]; but we will use Definition 5 for any values of m and n.

In what follows, we will assume that the structure does not imply rank deficiency. This is not really a restriction, since for a structure block implying rank deficiency, one can just remove a few rows or columns from the structure block until it does not imply rank deficiency anymore.

DEFINITION 6 (sparsity pattern). [illegible] $A \in \mathbb{C}^{m \times n}$ [illegible] $\mathcal{B} = (i, j, r)$ [illegible] $A = QR$ [illegible] sparsity pattern induced by the structure block $\mathcal{B}$. Q^H [illegible] $Q^H = Q_3^H Q_2^H Q_1^H$ [illegible]

- Q_1^H [illegible] $i, \ldots, m$ ([illegible] $\mathcal{B}$ [illegible] r [illegible])
- Q_2^H [illegible] $1, \ldots, i + r - 1$ ([illegible] $\mathcal{B}$)
- Q_3^H [illegible] $j + 1, \ldots, m$ ([illegible] $\mathcal{B}$)

[illegible] 3.1

Note that the above definition makes sense since if A fulfills a structure block $\mathcal{B}$, then A admits a QR-factorization satisfying the sparsity pattern induced by $\mathcal{B}$.

We should stress that the above definition was formulated from the point of view of a [illegible] structure block. In practical situations, there will probably be more than one structure block, causing each of the unitary components Q_i^H, $i = 1, 2, 3$, to have an additional decomposition into a sparse product of elementary unitary transformations. To stress this point, we indicated in Figure 3.1 the relation with the

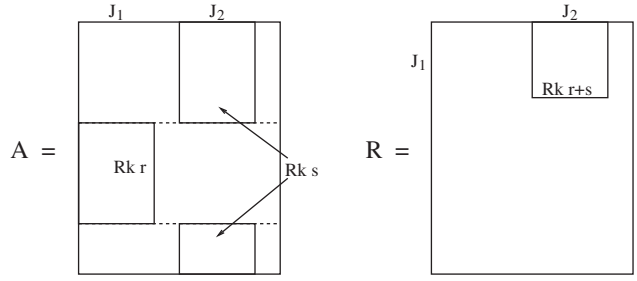


FIG. 3.2. *The figure shows the inheritance of structure by the R-factor for a general example. We note that the index sets I_1 and I_2 must be complementary to each other.*

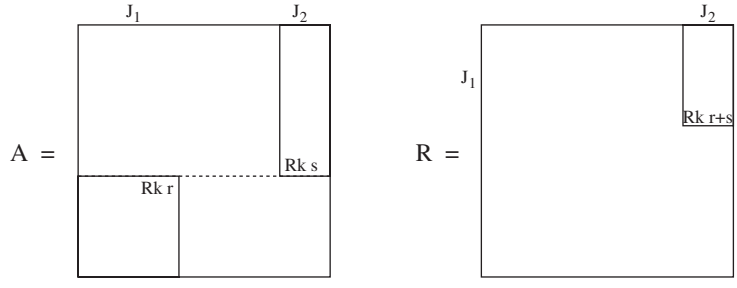


FIG. 3.3. *The figure shows the practical form in which we will exploit the inheritance of structure by the R-factor of Figure* 3.2. *In particular, it is assumed in this figure and in what follows that the index set J_2 takes the form $\{j_2, \ldots, n\}$ for certain j_2, since this is the practical form in which the inheritance result will be exploited.*

preparative and residual unitary transformations, corresponding to the upward- and downward-pointing arrows, respectively.

Since Definition 6 was formulated from the point of view of a [illegible] structure block, it does not reflect the complete sparsity pattern of the preparative and residual unitary operations. Nevertheless, it will be sufficiently strong for establishing the following result, which is a generalization of earlier results in the literature for some very special types of rank structures [8, 11].

THEOREM 7 (inheritance of structure by the R-factor). [illegible] $A \in \mathbb{C}^{m \times n}$ [illegible] $A(I_1, J_1) = \mathrm{Rk}\ r$ [illegible] $A(I_2, J_2) = \mathrm{Rk}\ s$ [illegible]

- I_1 [illegible] I_2 [illegible] $\{1, 2, \ldots, m\}$
- $J_1 = \{1, 2, \ldots, j_1\}$ [illegible] j_1
- J_2 [illegible]

[illegible] $A = QR$ [illegible] $R(J_1, J_2) = \mathrm{Rk}(r+s)$ ([illegible] 3.2) [illegible] A [illegible]

A [illegible] P [illegible] I_1 [illegible] I_2 [illegible] $\{1, 2, \ldots, i\}$ [illegible] $\{i+1, i+2, \ldots, m\}$ [illegible] i [illegible] 3.3 [illegible] $\mathrm{Rk}\ r$ [illegible] $\mathcal{B}$ [illegible] $PA = PQR$ [illegible] $\mathcal{B}$ [illegible]

. First consider the case where A is nonsingular. First, we claim that the Q-factor must "inherit" each of the Rk r low rank blocks situated entirely at the left border of A. Indeed, this is evident by using the equation $Q = AR^{-1}$, where the matrix R^{-1} takes linear combinations of the columns of A, hereby only involving "previous" columns, and hence not destroying such low rank blocks (see also [5]). This establishes the inheritance of structure by the Q-factor, i.e., $Q(I_1, J_1) = \text{Rk } r$. It follows that

$$\begin{aligned} R(J_1, J_2) &= Q^H(J_1, \{1, \ldots, m\})A(\{1, \ldots, m\}, J_2) \\ &= Q^H(J_1, I_1)A(I_1, J_2) + Q^H(J_1, I_2)A(I_2, J_2) \\ &= \text{Rk } r + \text{Rk } s \\ &= \text{Rk}(r + s), \end{aligned}$$

where the first transition follows from the QR-equation $R = Q^H A$, where the second transition follows from our assumption that I_1 and I_2 form a partition of $\{1, \ldots, m\}$, and where the third transition uses the fact that $Q(I_1, J_1) = \text{Rk } r$ and $A(I_2, J_2) = \text{Rk } s$.

Let us now consider the case where A is a general rectangular matrix. From our assumption that Q^H satisfies the sparsity pattern induced by the Rk r structure block, it can be shown that the equation $Q(I_1, J_1) = \text{Rk } r$ remains valid in this case (see also [4]), and hence one could use exactly the same proof as above. Alternatively, one could proceed by a direct argument; see Figure 3.4. □

One may ask where in Figure 3.4 the condition is used that the rank structure does not imply rank deficiency. This is done in Figure 3.4(b), where this condition guarantees that the created block of zeros is situated entirely in the strictly lower triangular part of the matrix. If this were not the case, then the index set indicated on the extreme left of Figure 3.4(c) would be a subset $\tilde{J}_1 \subset J_1$, so that Figure 3.4(d) would only lead to the weaker conclusion $R(\tilde{J}_1, J_2) = \text{Rk}(r + s)$.

For the remainder of this section, we turn to the practical exploitation of the above rank inheritance results for the Q- and R-factors of the QR-factorization.

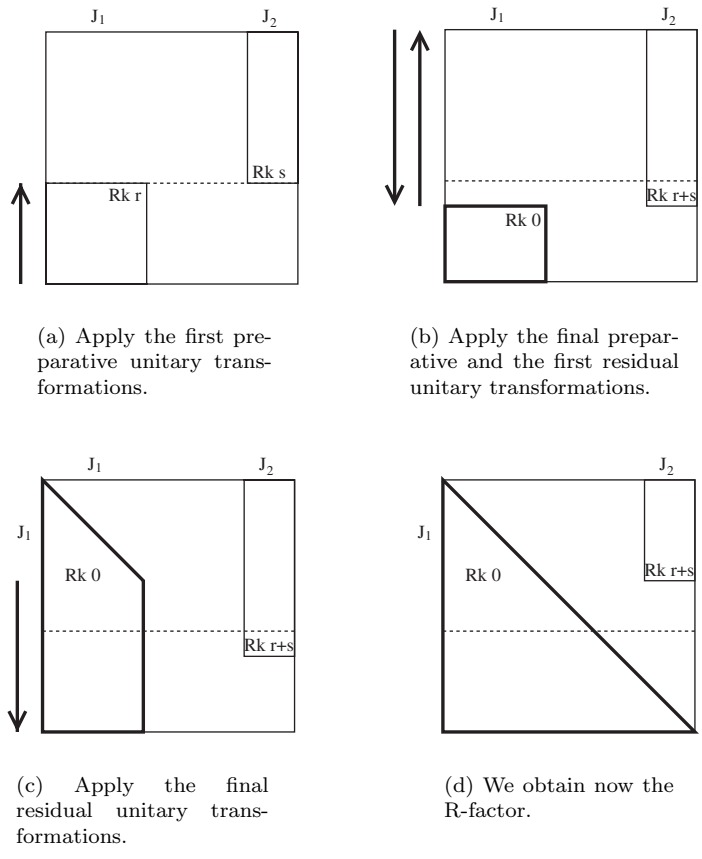
3.2. Algorithm for the preparative phase. In the next two subsections, we will work under the condition that A is a rank structured matrix for which a Givens-weight representation is available. More precisely, it will be assumed that the structured lower triangular part of A is represented by a -based Givens-weight representation, while the structured upper triangular part is represented by a - based Givens-weight representation; cf. section 2. Moreover, the structure blocks of the block lower triangular part are not allowed to intersect those of the block upper triangular part.

Given these input conditions, we will first describe an algorithm for the first phase of the QR-factorization, the so-called . We will illustrate the algorithm for a general type of rank structure, a slice of which is shown in Figure 3.5.

The corresponding Givens-weight representation is then shown in Figure 3.6.

Note that the subsequent unitary operations U_k, $k = K, \ldots, 1$, which will serve to compress the structure blocks in the lower triangular part are already in the Givens-weight representation, precisely by the concept of Givens-weight representation. Therefore, the algorithm will suffice with these operations from the representation, in a sense to be explained further.

During the algorithm, we are faced with the following problem: The Givens-weight representation is by definition an representation, based on a QR-factorization, where the weights were stored each time just at the moment when they would go beyond the top border of the structure (section 2). The problem is now that we want



(a) Apply the first preparative unitary transformations.

(b) Apply the final preparative and the first residual unitary transformations.

(c) Apply the final residual unitary transformations.

(d) We obtain now the R-factor.

FIG. 3.4. *The figure shows the inheritance of structure by the R-factor of Figure* 3.3 *using a direct argument in terms of the sparsity pattern of the matrix* Q^H.

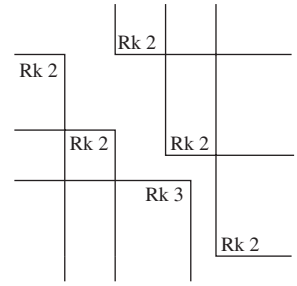


FIG. 3.5. *Starting rank structure.*

to apply the unitary transformations U_k, $k = K, \ldots, 1$, to the [illegible] matrix, hereby also updating the representation in the upper triangular part.

To update the structure in the upper triangular part of the matrix, the algorithm makes use of the general techniques for updating the Givens-weight representation under the influence of elementary unitary transformations described in [3] in the form of what we called there a [illegible]. The process is shown in Figure 3.7.

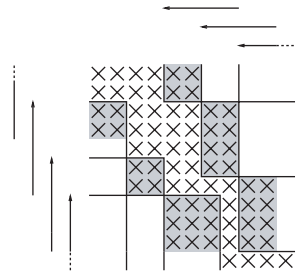
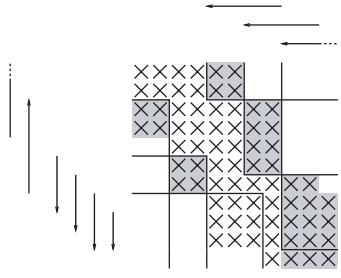
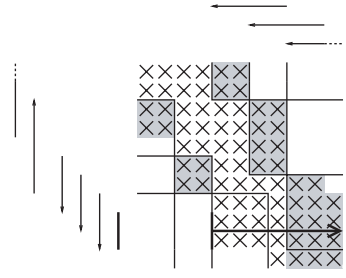


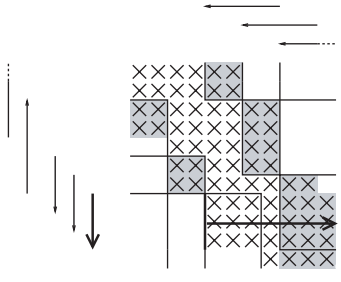
FIG. 3.6. *Starting Givens-weight representation for the rank structure in Figure* 3.5. *For clarity of the figure, the Givens transformations are shown grouped together as unitary operations.*



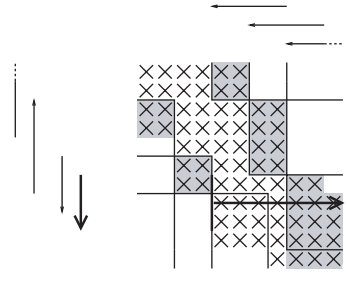
(a) Givens-weight representation after having reduced the Rk 3 block into a block of zeros, except for its top 3 rows.



(b) Apply the next Givens arrow. We apply it only to columns $5, 6, \ldots,$ since the result in columns $1, \ldots, 4$ is already available.



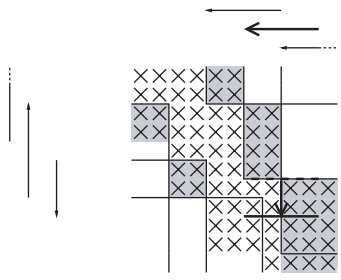
(c) Apply the second Givens arrow.



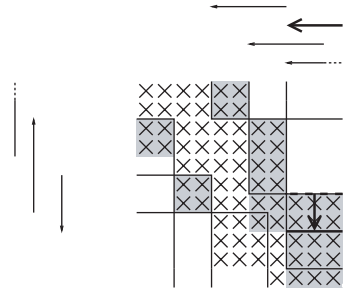
(d) Apply the third Givens arrow.

FIG. 3.7. *Preparative phase* (a)–(d).

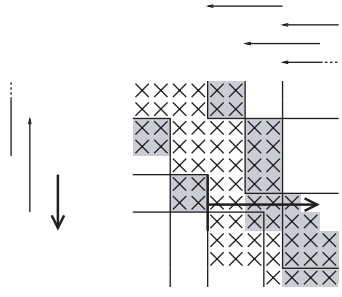
Let us comment on this figure. Figure 3.7(a) shows the starting Givens-weight representation. We assume here that the bottommost structure block $\mathcal{B}_{k+1}$ (say) has already been transformed into its zeroed form. This has resulted in the fact that the corresponding unitary compression operation U_{k+1}, corresponding to the thin upward pointing arrow acting on rows $8, 9, 10, 11$ in Figure 3.6, has already been peeled off, i.e., it has disappeared from the representation. Correspondingly, the weight block in rows $8, 9, 10$ and columns $5, 6, 7$ of the weight matrix in Figure 3.7(a) has turned from gray into white. The bottom right elements in Figure 3.7(a) are assumed to be disturbances coming from previous operations.



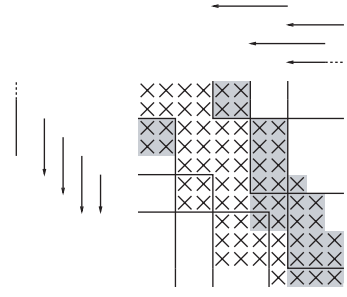
(e) Enlarge the action radius of the indicated unitary component $V_{l,1}$ from 6 to 8.



(f) Apply a unitary operation $V_{l,2}$ to bring the weights as much as possible to the left, and update $V_{l,1} := V_{l,1}V_{l,2}$.



(g) Apply the fourth Givens arrow.



(h) The current Rk 2 structure block has now been completely reduced into a block of zeros, except for its top 2 rows. In the figure, the next unitary operation is already shown as a decomposition of Givens arrows.

FIG. 3.7. *Preparative phase* (e)–(h).

We are then at the point of applying the next unitary compression operation U_k. Figure 3.7(a) shows this unitary operation U_k in the form of an explicit decomposition as a product of four downward-pointing Givens arrows.

Figures 3.7(b), 3.7(c), and 3.7(d) show the application of the Givens transformations belonging to the first three Givens arrows of the unitary compression operation U_k. Since part of their application is already available, by the concept of Givens-weight representation, these Givens arrows should be applied only to the columns lying strictly on the right of their current action radius, in this case columns $5, 6, \ldots$. These columns are indicated by the range of the bold horizontal arrow in Figures 3.7(b), 3.7(c), and 3.7(d).

In these figures, we used the following graphical code. The "active" Givens arrow which is currently being applied to the rows or columns of the matrix is always shown in boldface. This may be both a Givens arrow belonging already to the Givens-weight

representation (as in the present case) or a new Givens arrow or unitary operation coming from outside (see further). On the other hand, Givens arrows or unitary operations belonging to the Givens-weight representation, but which are not active in the current step of the algorithm, are always represented by [illegible] arrows.

For example, the operation in Figure 3.7(b) can be expressed in MATLAB notation as

$$W([9:10],[5:11]) := GW([9:10],[5:11]),$$

where G denotes the applied Givens arrow (consisting in this case only of a single Givens transformation), and where W denotes the weight matrix of the Givens-weight representation. Similar expressions hold for the operations in Figures 3.7(c) and 3.7(d).

We should still explain why it is valid to apply the row operations directly to the weights of the upper triangular representation, as in columns $9, 10, 11$ of Figures 3.7(b), 3.7(c), and 3.7(d). To this end, we recall that the weights contain a kind of compressed information about the matrix, and that in order to obtain these elements in full form, the weights should first be spread out by the unitary column operations of the Givens-weight representation of the upper triangular part. But clearly, by the associativity of matrix multiplication, it does not matter whether we first spread out the weights by the use of these [illegible] operations, or instead first apply the disturbing [illegible] operations. This shows that, indeed, it is correct to apply the row operations directly to the weights.

We are now at the point of applying also the fourth Givens arrow of U_k to the rows. But the application of this fourth Givens arrow would lead to a mix of real-size elements and weights in the submatrix $W([6:8],[7:8])$, which is definitely not correct.

The solution to this problem consists in "enlarging" the column representation. This means that we bring the two rows lying just below the new structure block $\mathcal{B}_l$, whose bottom leftmost element has coordinates $(6,7)$ in Figure 3.7(d), "into" the column representation. Practically, this is achieved by [illegible] of the corresponding unitary column operation of the Givens-weight representation of the upper triangular part (let us call this operation $V_{l,1}$); i.e., we should apply $V_{l,1}$ to all rows between its present and its new action radius, in the present case rows 7, 8. See Figure 3.7(e).

The operation in Figure 3.7(e) can be expressed in MATLAB notation as

$$W([7:8],[7:10]) := W([7:8],[7:10])V_{l,1}.$$

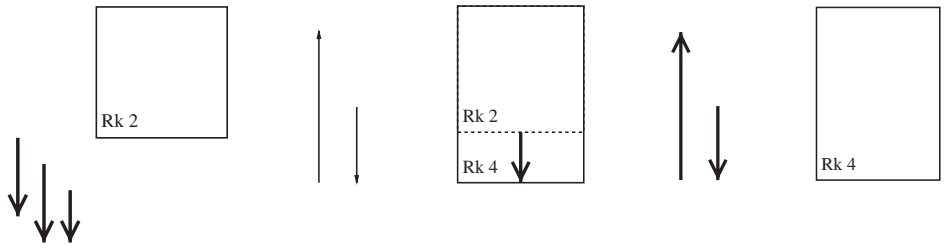
Having enlarged the action radius of $V_{l,1}$, it is now safe to apply the fourth Givens arrow of U_k to the rows. Before doing this, however, we note that applying all these row operations would ultimately lead to a complete fill-in in the upper triangular part of the weight matrix. Since we want to minimize this fill-in as far as possible, we first apply an auxiliary unitary compression transformation $V_{l,2}$ to the columns, in order to bring the newly introduced weights as far as possible to the left. See Figure 3.7(f).

The operation in Figure 3.7(f) can be expressed in MATLAB notation as

$$W([7:8],[9:11]) := W([7:8],[9:11])V_{l,2},$$

where $V_{l,2}$ is chosen such that $W([7:8],[9:11])V_{l,2}$ takes a lower triangular form.

Note that this auxiliary operation $V_{l,2}$ is really new in the sense that it was not present yet in the original Givens-weight representation. Moreover, we apply it only



(a) Apply the first Givens arrows. This corresponds to Figures 3.7(b), 3.7(c), and 3.7(d).

(b) Add two new rows to the structure block. This corresponds to Figure 3.7(e).

(c) Apply the next Givens arrow and unitary operation. This corresponds to Figure 3.7(g).

FIG. 3.8. *Mechanism underlying the structure block movement in the preparative phase.*

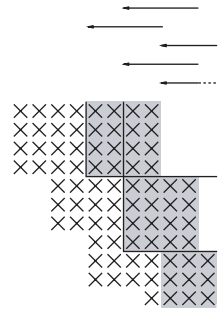


FIG. 3.9. *Givens-weight representation after the complete preparative phase.*

to the upper triangular rank structured part of the matrix, in the present case rows 7, 8. This means that $V_{l,2}$ is an "internal" operation, which should be concatenated to the Givens-weight representation of the upper triangular part; i.e., we should update the unitary component as $V_l := V_{l,1}V_{l,2}$.

Having done all these preparations, we can finally apply the fourth Givens arrow of U_k to the rows; see Figure 3.7(g). We have then completely finished the Givens arrows belonging to the current unitary transformation U_k. Note that the corresponding weight block, lying on the intersection of rows 6, 7 and columns 3, 4 of Figure 3.7(h), has turned from gray into white: This indicates the fact that these elements no longer contain any "encoded" information, but that they contain precisely the real-size elements standing there at this particular moment of the algorithm. The reason underlying this is nothing but the concept of Givens-weight representation.

At this moment, we are at the point of embarking the Givens arrows belonging to the next unitary operation U_{k-1}. Because the situation in Figure 3.7(h) is similar to the one we started from in Figure 3.7(b), this process is not shown.

Figure 3.8 summarizes the explained mechanism of the preparative phase in terms of the structure blocks in the structured upper triangular part. Note that we use here the same graphical code as in the previous figures; i.e., in each step, only the boldfaced vertical line segments are actually applied to the rows of the matrix.

Figure 3.9 shows the final situation for the example in Figure 3.7 at the end of the complete preparative phase. Note that the column representation has "grown," corresponding to the fact that the ranks in the upper triangular part have increased, which is consistent with Figure 3.8.

It can also be seen from this figure that the representation for the block lower triangular part has completely been peeled off, i.e., it has [illegible]. This underlies the fact that the structure blocks in the block lower triangular part have been transformed into their zeroed form.

In order to make the matrix completely upper triangular, we should then still remove a few subdiagonals. The process of doing this will be the subject of the next subsection.

3.3. Algorithm for the residual phase. In this subsection, we show that by using the Givens-weight representation, the second phase of the QR-factorization can also be performed in an efficient way, the so-called [illegible].

We will explain the algorithm for the Givens-weight representation shown in Figure 3.9.

The application of the residual phase makes use of the general techniques for updating the Givens-weight representation under the influence of Givens transformations described in [3] in the form of what we called there a [illegible]. This process is shown in Figure 3.10.

Let us comment on Figure 3.10. The basic flow of the algorithm is determined by applying a sequence of upward-pointing Givens arrows making the subsequent columns k upper triangular, $k = 1, \ldots, n-1$.

Figure 3.10(a) shows the application of the Givens arrow $G_{1,2}G_{2,3}G_{3,4}$ making the first column upper triangular.

The action of Figure 3.10(a) can be expressed in MATLAB notation as

$$W([1:4],[1:8]) = G_{1,2}G_{2,3}G_{3,4}W([1:4],[1:8]),$$

where the Givens transformations $G_{i,i+1}$, $i = 3, 2, 1$, are chosen such that the column vector $G_{1,2}G_{2,3}G_{3,4}W([1:4],1)$ is brought in upper triangular form.

Figure 3.10(b) shows the application of the Givens arrow making the second column upper triangular.

We would then like to apply the Givens arrow making the third column upper triangular. But we should be careful that there is no mixture of real-size elements and weights in the weight matrix during this process. Therefore, before making the third column upper triangular, we first have to [illegible]; i.e., we first have to regress the action radius of the unitary component $V_l = V_{l,1}V_{l,2}$ highlighted in Figure 3.10(c). We regress here from row 4 down to row 2, since we are intending to apply in the next step an operation acting on rows $3, \ldots, 7$.

The action of Figure 3.10(c) can be expressed in MATLAB notation as

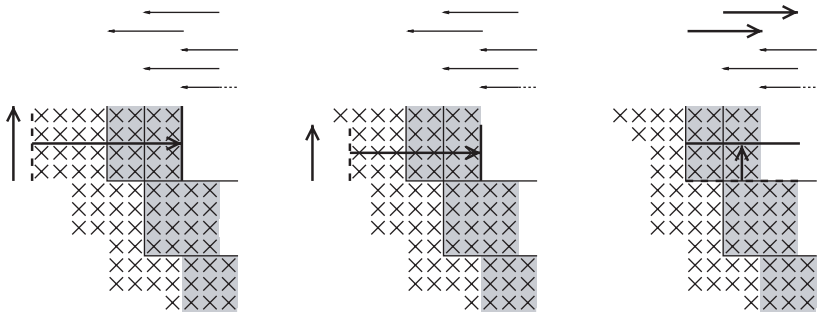
$$W([3:4],[5:8]) = W([3:4],[5:8])V_l^H.$$

Having done this regression operation, the next two columns are made upper triangular in Figures 3.10(d) and 3.10(e).

At this moment, we are at the point of embarking the following columns and making them upper triangular. Because this problem is similar to the one from which we started in Figure 3.10(a), this process is not shown.

Figure 3.11 summarizes the explained mechanism of the residual phase in terms of the structure blocks in the upper triangular part.

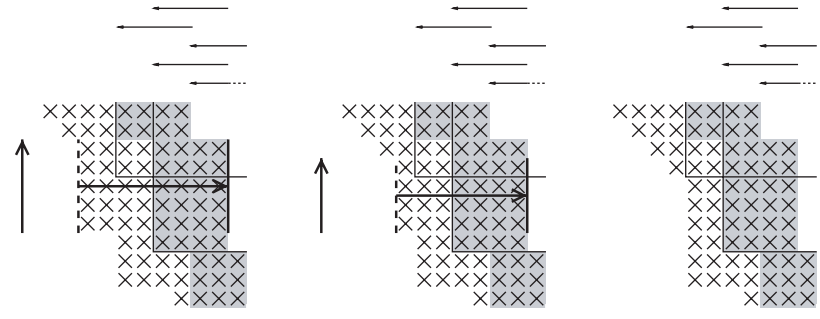
Figure 3.12 shows the final situation at the end of the complete residual phase. Note that the column representation has "lost" some terrain in the sense that it has regressed to the direction of the top right corner of the matrix. This is consistent



(a) Bring the first column in upper triangular form.

(b) Bring the second column in upper triangular form.

(c) Regress the action radius of the indicated unitary operation $V_l = V_{l,1}V_{l,2}$ from row 4 to row 2.



(d) Bring the third column in upper triangular form.

(e) Bring the fourth column in upper triangular form.

(f) The first four columns are now upper triangular.

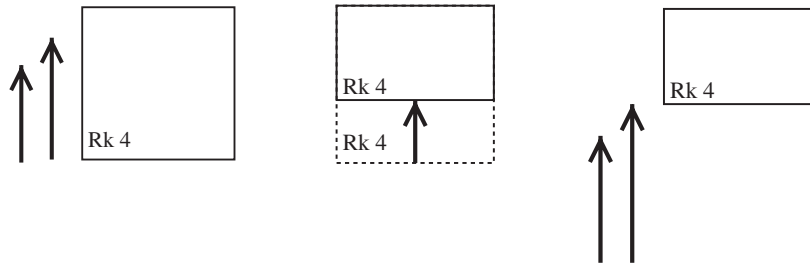
FIG. 3.10. *Residual phase.*

with the mechanism illustrated in Figure 3.11. On the other hand, note that the unitary transformations involved in the Givens-weight representation for the block upper triangular part have remained exactly the same. This underlies the fact that the ranks and the induced column dependencies have been left unchanged under the regression process.

For a global overview, the reader could also have a second look at the original algorithm flow which was sketched in Figure 3.4.

Let us note that the above procedure for the residual phase guarantees that the structure blocks of R are all lying in the strictly upper triangular part of this matrix. In a certain sense, this may seem to conflict with Figure 3.3, which predicts that in certain cases, the structure blocks of R could reach beyond the main diagonal. In fact, the above algorithm will have performed an implicit [illegible] of structure in this case. But this did not occur for the example which we have chosen.

Summarizing, by the algorithm of the current section, we have obtained a QR-factorization $A = QR$, where the Q-factor is decomposed as a product of Givens transformations, and where the R-factor is represented by a column-based Givens-



(a) Make the first columns upper triangular. This corresponds to Figures 3.10(a) and 3.10(b).

(b) Remove two rows from the structure block. This corresponds to Figure 3.10(c).

(c) Make the next columns upper triangular. This corresponds to Figures 3.10(d) and 3.10(e).

FIG. 3.11. *Mechanism underlying the structure block movement in the residual phase.*

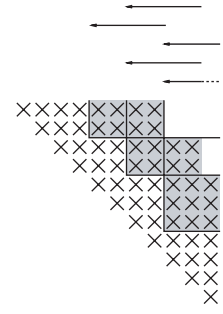


FIG. 3.12. *Givens-weight representation after the complete residual phase. The reader should check that the underlying structure blocks correspond with the predictions in Theorem* 7.

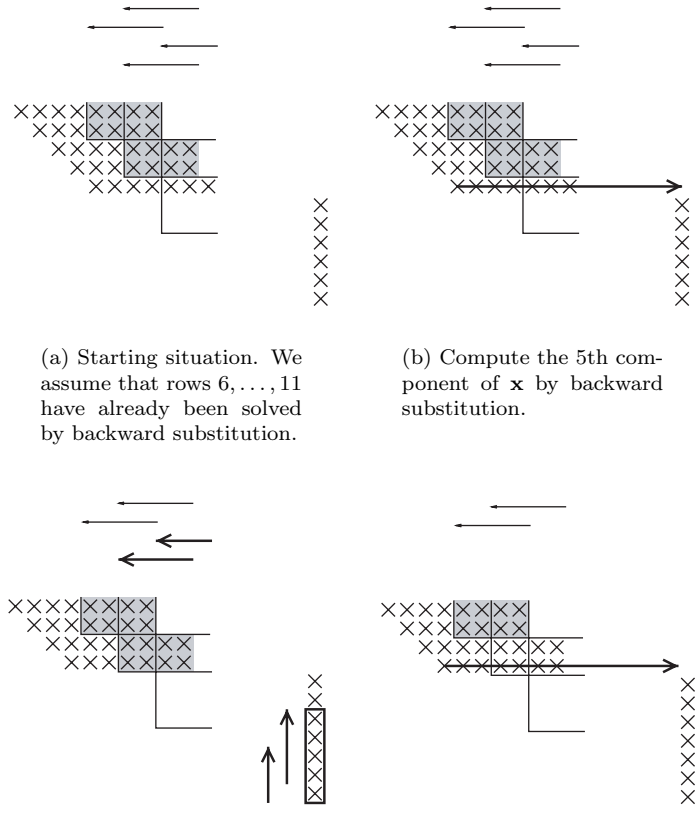
weight representation. In the next section, we will show how this QR-factorization can be used for the solution of a linear system.

. . . 8 (algorithm complexity). For the types of rank structures occurring in practice, the complexity of the preparative and residual phase can be estimated as follows. Assume that the starting Givens-weight representation was efficient in the sense that it consists of a number of $O(rn)$ Givens transformations, where n is the matrix size and r is some measure for the average rank of the rank structure. Denote with s some measure for the bandwidth of the unstructured matrix part around the main diagonal. Then in both the preparative and residual phase, we apply a number of $O(rn) + O(sn)$ Givens transformations to the rows of the matrix A. Moreover, each of these Givens transformations acts on a number of approximately $O(r) + O(s)$ elements of the weight matrix. It follows that the global algorithm complexity for the QR-factorization equals $O((ar^2 + brs + cs^2)n)$ operations, where $a, b, c \in \mathbb{R}$ are certain weighting parameters.

4. Solution of a linear system. In this section we shall use the QR-factorization to solve a linear system $A\mathbf{x} = \mathbf{b}$. We do this by rewriting the linear system in the form

$$R\mathbf{x} = Q^H\mathbf{b} =: \tilde{\mathbf{b}}, \tag{4.1}$$

which can then be solved by backward substitution; see Figure 4.1.



(a) Starting situation. We assume that rows $6, \ldots, 11$ have already been solved by backward substitution.

(b) Compute the 5th component of $\mathbf{x}$ by backward substitution.

(c) Apply available unitary operations V_l to compress the matrix R. (No actual operations have to be performed.) Multiply (a working copy of) the vector $\mathbf{x}$ with the inverse operations.

(d) We can now go on to compute the 4th component of $\mathbf{x}$ by backward substitution.

FIG. 4.1. *Direct solution of a linear system.*

Let us comment on this figure. The basic flow of the algorithm is determined by solving the subsequent rows of the linear system (4.1) by backward substitution, hereby obtaining the subsequent components of the indeterminate vector $\mathbf{x}$.

Figure 4.1(a) shows the starting situation. We assume here that rows $11, \ldots, 6$ of the linear system (4.1) have already been solved by backward substitution, thereby obtaining the corresponding components $x_{11}, \ldots, x_6$ of the vector $\mathbf{x}$. In the right part of the figure, we use a column of vertical crosses to denote these already computed components of $\mathbf{x}$. Note also that the elements in rows $11, \ldots, 6$ of the weight matrix have been dropped from the figure. We do this as an intuitive means of indicating that these elements will no longer be needed during the rest of the algorithm.

We can now solve the 5th row of the linear system (4.1), hereby obtaining the component x_5. This is indicated pictorially in Figure 4.1(b). Note that the arrow

in this figure serves as an intuitive means to denote that the 5th component of $\mathbf{x}$ is computed from the information in the elements in the 5th row of the weight matrix, hereby explaining the rightward direction of this arrow.

Using MATLAB notation, the action in Figure 4.1(b) can be expressed as

$$\mathbf{x}(5) := \frac{1}{W(5,5)}(\tilde{\mathbf{b}}(5) - W(5,[6:11])\mathbf{x}([6:11])).$$

Having performed the action in Figure 4.1(b), the elements in the 5th row of the weight matrix will no longer be needed, and so we can drop them from the figure. We would then like to compute the next component x_4. But then we are going to enter a new structure block $\mathcal{B}_l$ in the upper triangular part of the weight matrix. Since we would like to have the coefficient matrix of our linear system as sparse as possible, this is now a good moment to multiply the upper triangular matrix R by means of the available unitary column operation $V_l = V_{l,1}V_{l,2}$ associated to this structure block $\mathcal{B}_l$, hereby compressing the structure of this matrix. Although this compression may sound rather expensive, from the computational point of view, nothing has to be done since the effect of this unitary operation V_l is already [illegible] by the concept of the Givens-weight representation. (In principle, we should still apply V_l to the rows below its current action radius, in the present case to the elements in rows $5, 6, \ldots$. But this is not necessary since the elements in rows $5, 6, \ldots$ have no function anymore during the rest of the algorithm.)

In other words, the compression operation of the matrix R shown in the left part of Figure 4.1(c) serves only for understanding the algorithm, but does not require any actual operation.

We have now modified the linear system (4.1) as

$$R\mathbf{x} = (RV_l)(V_l^{-1}\mathbf{x}) = \tilde{\mathbf{b}}.$$

Thus the indeterminate vector $\mathbf{x}$ should be multiplied with the inverse of the unitary operation V_l. (Of course, in order not to overwrite the already computed values of $\mathbf{x}$, one should use a working copy $\tilde{\mathbf{x}}$ of the vector $\mathbf{x}$ on which to perform these operations!) The latter operations are indicated by the fat vertical arrows shown in the right part of Figure 4.1(c).

Using MATLAB notation, the action in Figure 4.1(c) can be expressed as

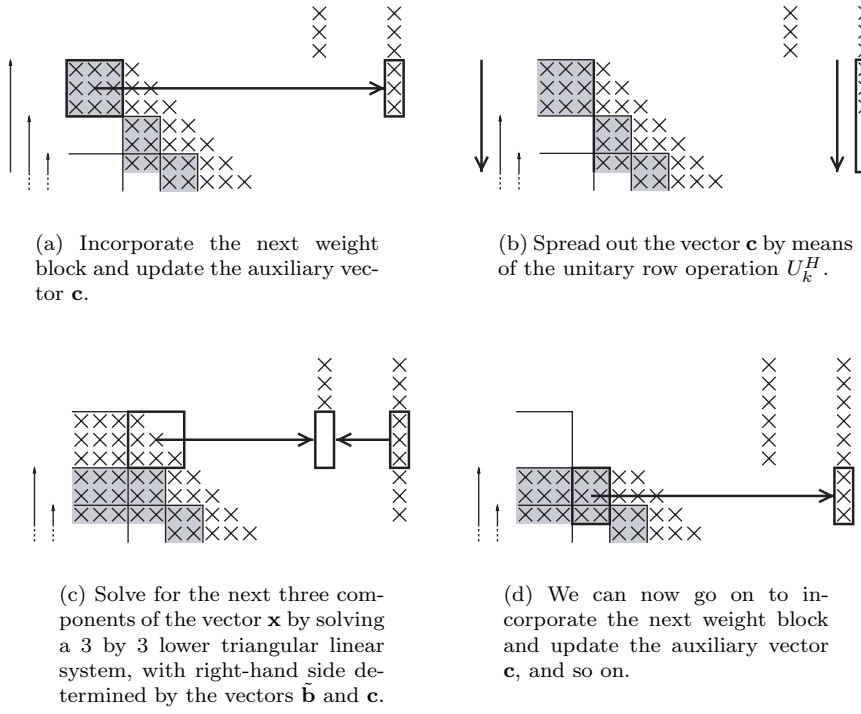
$$\mathbf{x}([7:11]) = V_l^H \mathbf{x}([7:11]),$$

where, as we already remarked, the already computed components of $\mathbf{x}$ should first be stored in a safe place to avoid losing them.

We can then go on to compute the next two components x_4, x_3 of the indeterminate vector by backward substitution. Since the situation in Figure 4.1(d) is similar to the one in Figure 4.1(b), these operations are not shown.

At the end of this process we will have obtained the full indeterminate vector $\mathbf{x}$, hereby solving the linear system.

For completeness of this paper, let us now describe a similar solution algorithm in cases where the R-factor is described by a [illegible] representation. Actually, we prefer to explain the algorithm in terms of a [illegible] matrix L (which is an equivalent problem since we could rewrite $R\mathbf{x} = \tilde{\mathbf{b}}$ as $LJ\mathbf{x} = J\tilde{\mathbf{b}}$, where $L := JRJ$, and with J the antidiagonal matrix; the row-based Givens-weight representation for R transforms in this way into a row-based Givens-weight representation for the matrix L).



(a) Incorporate the next weight block and update the auxiliary vector **c**.

(b) Spread out the vector **c** by means of the unitary row operation U_k^H.

(c) Solve for the next three components of the vector **x** by solving a 3 by 3 lower triangular linear system, with right-hand side determined by the vectors $\tilde{\mathbf{b}}$ and **c**.

(d) We can now go on to incorporate the next weight block and update the auxiliary vector **c**, and so on.

FIG. 4.2. *Direct solution of a linear system in the case of a row-based Givens-weight representation.*

Thus we will solve a linear system of the form

$$L\mathbf{x} = \tilde{\mathbf{b}}. \tag{4.2}$$

This will be achieved by computing the subsequent components $x_1, x_2, \ldots$ of **x** by forward substitution. The algorithm is explained in Figure 4.2.

Let us comment on this figure. Figure 4.2(a) shows the starting situation, where it is assumed that the first three components, x_1, x_2, x_3, have already been computed by forward substitution. These components are depicted by the vertical sequence consisting of three crosses in the top right part of the figure. The elements in the first three rows of the weight matrix will no longer be needed during the rest of the algorithm, and therefore they have been dropped from the figure.

Note that the top right part of Figure 4.2(a) shows also a second vector **c**, of which already six components have been computed. This will be an auxiliary vector. It contains the matrix-vector product of the already computed components of **x** with the structured lower triangular matrix part of the lower triangular matrix L.

Figure 4.2(a) shows how to update this auxiliary vector **c** by incorporating the indicated weight block of the weight matrix and multiplying it with the first three components of **x**. This contribution is then added to the vector **c**.

In MATLAB notation, the operation in Figure 4.2(a) corresponds to updating

$$\mathbf{c}([4:6]) = \mathbf{c}([4:6]) + W([4:6],[1:3])\mathbf{x}([1:3]).$$

The left part of Figure 4.2(b) shows how we spread out the weight matrix by means of the next unitary component U_k^H in order to obtain the real-size form of

the elements in rows $4, 5, 6$ of the lower triangular matrix L. These spreading-out operations serve only as an aid for understanding the algorithm, but they are not actually computed. What we do perform is spreading out the vector $\mathbf{c}$ by means of this same operation U_k^H, as illustrated in the right part of Figure 4.2(b).

In MATLAB notation, the operation in Figure 4.2(b) corresponds to updating

$$\mathbf{c}([4:9]) = U_k^H \mathbf{c}([4:9]).$$

Having performed this spreading-out operation, the elements in rows $4, 5, 6$ of the weight matrix in Figure 4.2(c) now contain the corresponding real-size elements of the lower triangular matrix L. Therefore, these rows can now be used to solve for the components x_4, x_5, x_6 of the indeterminate vector. The coefficient matrix of this linear system is given by the lower triangular 3 by 3 matrix surrounded by the thick black box in Figure 4.2(c). The right-hand side of the linear system is determined by the actual right-hand side vector $\tilde{\mathbf{b}}$, from which is subtracted the contribution of the matrix-vector product of the already computed components of $\mathbf{x}$ with the structured lower triangular matrix part, which is contained in the vector $\mathbf{c}$.

In MATLAB notation, the operation in Figure 4.2(c) corresponds to setting

$$\mathbf{x}([4:6]) = W([4:6],[4:6])^{-1}(\tilde{\mathbf{b}}([4:6]) - \mathbf{c}([4:6])).$$

Having computed the components x_4, x_5, x_6, the elements in rows $4, 5, 6$ of the weight matrix can now be dropped from the next figures, starting with Figure 4.2(d). We can then move on to the next weight block, which is shown boxed in Figure 4.2(d), and use it to update the vector $\mathbf{c}$:

$$\mathbf{c}([7:9]) = \mathbf{c}([7:9]) + W([7:9],[4:5])\mathbf{x}([4:5]).$$

The next operations are not shown.

We note that the flow of this row-based algorithm is very similar to that for block quasi-separable matrices, first reported in [8]. On the other hand, the column-based algorithm described earlier in this section does not seem to have such an interpretation.

. . . 9 (overdetermined systems). In this and the previous section, we implicitly assumed that the coefficient matrix A, and hence the R-factor of its QR-factorization, were , matrices. But this condition is irrelevant: Also in the case of a (full-rank) overdetermined linear system with $A \in \mathbb{C}^{m \times n}$ and $m \geq n$, one can compute the QR-factorization and the corresponding least-squares solution to a linear system in exactly the same way as before.

. . . 10 (algorithm complexity). Note that in the above algorithms for the solution of a linear system, each parameter of the Givens-weight representation is used , during the algorithm, each time counting for $O(1)$ floating point operations. Thus if the starting Givens-weight representation was efficient, it follows that the global algorithm complexity for the solution of the linear system equals $O((dr + es)n)$ operations for suitable $d, e \in \mathbb{R}$. Compare with Remark 8.

5. Numerical experiments. To check the accuracy and the numerical stability of the algorithms in computing the QR-factorization and solving the corresponding linear system based on this factorization, we have performed several numerical experiments. The algorithms were implemented in MATLAB. The experiments were executed on an Intel PC running MATLAB Version 7.0.1.24704 (R14) under Linux having 1 GByte of memory and an Intel Pentium 4 processor running at 3.2 GHz. The software of these numerical experiments can be requested from the authors.

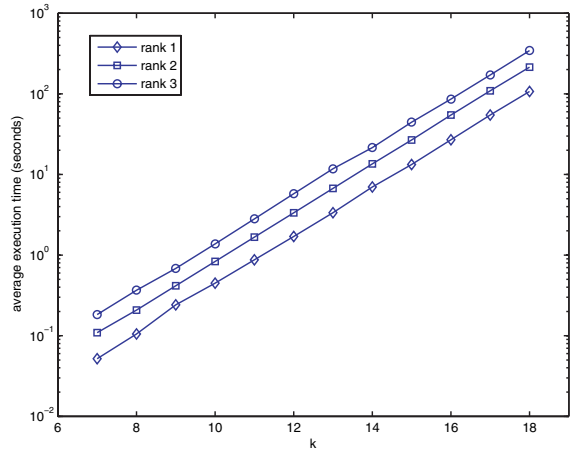


FIG. 5.1. *Average execution time for* 5 *random samples of size* $n = 2^k$ *and rank* $r = 1, 2, 3$.

Experiment 1. We constructed nonsymmetric rank structured matrices of sizes n by n, with $n = 2^k$ for $k = 7, \ldots, 18$. The structure blocks were situated just below the main diagonal, following immediately one after the other, and all having the same rank index r. Formally, this means that $\mathcal{B}_k : (i_k, j_k, r_k) = (k+1, k, r)$, $k = 1, \ldots, n-1$. The upper triangular part was rank structured in the same way as the lower triangular part; in particular, the bandwidth of the rank unstructured matrix part s was equal to one. To guarantee working with a Givens-weight representation consisting of only $O(rn)$ parameters, we made use of the canonical Givens-weight representation "of type 1" described in [3].

For each matrix size n, the rank indices $r = 1, 2, 3$ were taken. For each of these sizes n and each of these rank indices r, 5 samples were considered. The right-hand side $\mathbf{b} \in \mathbb{C}^n$ of each linear system $A\mathbf{x} = \mathbf{b}$ was generated with entries uniformly random distributed between 0 and 1.

Figure 5.1 shows for each size $n = 2^k$ and each rank index r the execution time $T_{k,r}$ averaged over the 5 samples of computing the QR-factorization and solving the corresponding linear system.

To check that the computational complexity is linear in the size n of the matrix, Figure 5.2 shows the fraction $T_{k+1,r}/T_{k,r}$ averaged over the 5 samples and over the ranks $r = 1, 2, 3$.

Figure 5.3 gives the average percentage of the total execution time spent to solve the linear system. It is seen from this figure that this takes only about 15% of the total execution time. In other words, once the QR-factorization has been computed as described in section 3, the subsequent solution of the linear system as described in section 4 is a relatively cheap process. This distinction becomes even more pronounced in case of higher ranks, since the QR-factorization has complexity $O(r^2 n)$, whereas the actual solver has complexity only $O(rn)$ (cf. Remark 10).

To check the accuracy of the algorithm, we considered matrices having singular values equidistant between 10^{-1} and 1; i.e., the condition number of each of the 5 samples is equal to 10. In the same way we took for each size n and each rank r 5 samples having condition number 10^{10}. To measure the accuracy, we computed the relative residual norm

$$\frac{\|A\mathbf{x} - \mathbf{b}\|}{\|A\|\|\mathbf{x}\| + \|\mathbf{b}\|}. \tag{5.1}$$

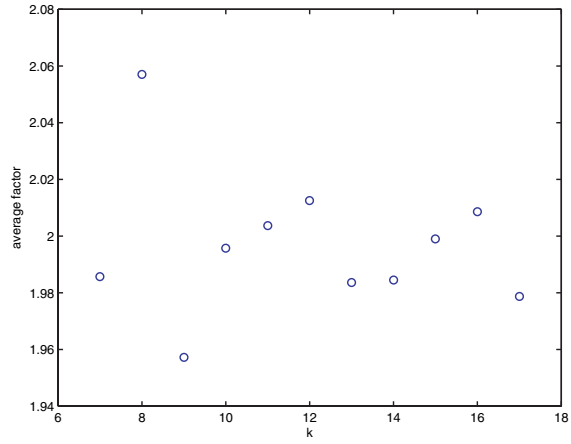


FIG. 5.2. *Fraction $T_{k+1,r}/T_{k,r}$ averaged over 5 random samples and over ranks $r = 1, 2, 3$ in function of size $n = 2^k$.*

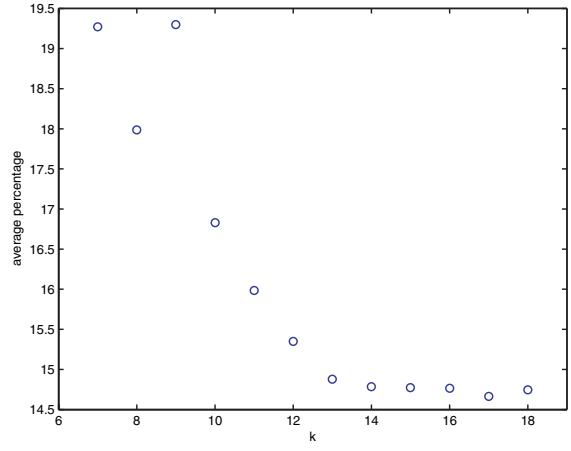


FIG. 5.3. *Percentage of total computing time spent to solve the linear system averaged over 5 random samples and over ranks $r = 1, 2, 3$ in function of size $n = 2^k$.*

(We evaluated the numerator in an efficient way by means of the algorithm for matrix-vector multiplication with a Givens-weight representation described in [3].) Figure 5.4 shows the relative residual norm averaged over the 5 samples and the ranks $r = 1, 2, 3$ for each of the condition numbers and each of the sizes.

From this figure, one can note the odd fact that the algorithm appears to be more stable for ill-conditioned than for well-conditioned linear systems. This is probably an artifact of the measure of stability that we used: It can be imagined that for ill-conditioned matrices with geometrically distributed singular values, the bound (5.1) is too pessimistic, in the sense that the denominator could grow faster than the numerator.

Finally, let us give some more details on the construction of the above test matrices. Starting from a diagonal matrix containing the desired singular values, we applied to rows and columns a "disturbing" sequence of Givens arrows of width r. This resulted in a nonsymmetric matrix having the required rank structure in both

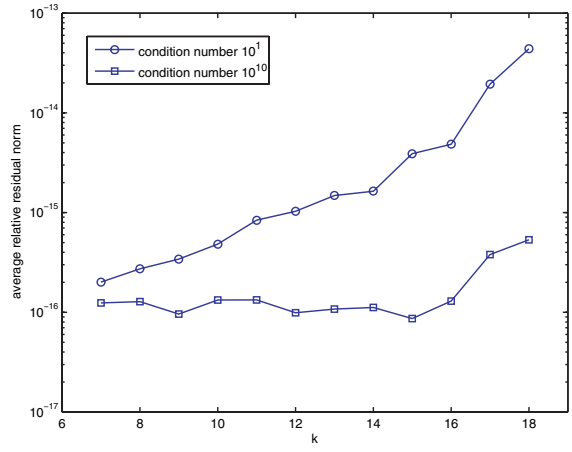


FIG. 5.4. *Relative residual norm averaged over* 5 *random samples and over ranks* $r = 1, 2, 3$ *in function of size* $n = 2^k$ *and condition number* 10^1 *and* 10^{10}.

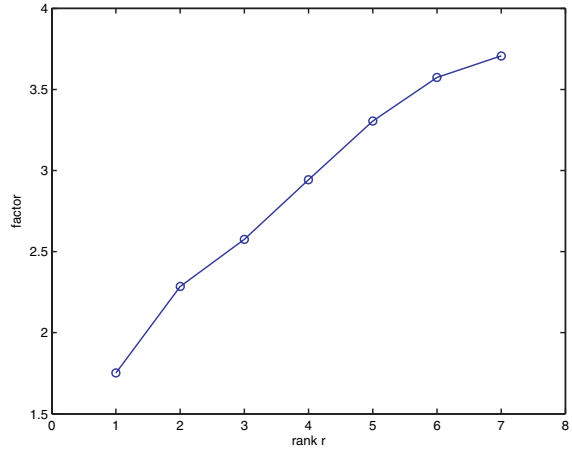


FIG. 5.5. *Fraction* T_{2r}/T_r *in function of the rank index* $r = 2^l$ *with* $l = 1, 2, \ldots, 7$.

lower and upper triangular parts.

Since it can be argued that the above construction yields rather "special" rank structured matrices, we next applied a "randomization" procedure. We did this by applying Givens transformations to rows and columns in such a way that both the lower and the upper rank structures of the matrix were preserved. Let us point out that this randomization procedure took about 95% of our total execution time. A detailed description of this perturbation method will not be given here. Moreover, we note that the results are quite the same with and without this perturbation approach.

Experiment 2. To check the computational complexity as a function of the rank index r, we considered the execution time T_r for matrices of fixed size $n = 2^{10} = 1024$ and varying rank index $r = 2^l$ with $l = 1, 2, \ldots, 8$. The actual construction of the test matrices was performed in exactly the same way as before. Figure 5.5 gives the fraction T_{2r}/T_r for subsequent ranks. Note that the fraction tends to approximate 4 for large rank indices r but is much smaller for small values of r. This value of 4

agrees with the fact that the starting Givens-weight representation consists of $O(rn)$ parameters, in which case the updating processes in section 3 both have an $O(r^2n)$ complexity (cf. Remark 8).

Finally, let us mention that the methods of this paper have been tested for other, more irregular examples of rank structures as well. The results are similar to those above.

6. Conclusion. In this paper we described an algorithm for performing the QR-factorization of a rank structured matrix using the Givens-weight representation. We showed how this QR-factorization could be used as a first step for solving linear systems. We described the underlying propagation of rank structure during the algorithm. The numerical performance of the algorithm was demonstrated by means of some numerical experiments.

REFERENCES

[1] S. Chandrasekaran and M. Gu, *Fast and stable algorithms for banded plus semiseparable systems of linear equations*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 373–384.

[2] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, and A.-J. van der Veen, *Fast stable solver for sequentially semi-separable linear systems of equations*, in High Performance Computing–HiPC 2002, Lecture Notes in Comput. Sci. 2552, Springer, Berlin, 2002, pp. 545–554.

[3] S. Delvaux and M. Van Barel, *A Givens-Weight Representation for Rank Structured Matrices*, Technical report TW453, Department of Computer Science, Katholieke Universiteit Leuven, Leuven (Heverlee), Belgium, 2006.

[4] S. Delvaux and M. Van Barel, *Rank structures preserved by the QR-algorithm: The singular case*, J. Comput. Appl. Math., 189 (2006), pp. 157–178.

[5] S. Delvaux and M. Van Barel, *Structures preserved by the QR-algorithm*, J. Comput. Appl. Math., 187 (2006), pp. 29–40.

[6] P. Dewilde and A.-J. van der Veen, *Time-Varying Systems and Computations*, Kluwer Academic Publishers, Boston, 1998.

[7] Y. Eidelman and I. C. Gohberg, *Fast inversion algorithms for a class of block structured matrices*, Contemp. Math., 281 (2001), pp. 17–38.

[8] Y. Eidelman and I. C. Gohberg, *A modification of the Dewilde-van der Veen method for inversion of finite structured matrices*, Linear Algebra Appl., 343–344 (2002), pp. 419–450.

[9] Y. Eidelman and I. C. Gohberg, *Fast inversion algorithms for a class of structured operator matrices*, Linear Algebra Appl., 371 (2003), pp. 153–190.

[10] I. C. Gohberg, T. Kailath, and I. Koltracht, *Linear complexity algorithms for semiseparable matrices*, Integral Equations Operator Theory, 8 (1985), pp. 780–804.

[11] E. Van Camp, N. Mastronardi, and M. Van Barel, *Two fast algorithms for solving diagonal-plus-semiseparable linear systems*, J. Comput. Appl. Math., 164–165 (2004), pp. 731–747.

[12] R. Vandebril, M. Van Barel, and N. Mastronardi, *A note on the representation and definition of semiseparable matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 839–858.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 491–496

A NOTE ON GENERIC KRONECKER ORBITS OF MATRIX PENCILS WITH FIXED RANK*

FERNANDO DE TERÁN† AND FROILÁN M. DOPICO†

Abstract. The set of $m \times n$ complex matrix pencils with rank (normal rank) at most r defines a subset of pencils in a complex $2mn$ dimensional space. For $r = 1, \ldots, \min\{m, n\} - 1$, we show that this subset is a closed set, which is the union of $r+1$ irreducible components. Each of these irreducible components is the closure of a certain orbit of strictly equivalent pencils with rank r. The Kronecker canonical forms of these orbits are explicitly described, and their dimensions are counted. These are the Kronecker canonical forms of generic pencils of rank at most r. If $m \neq n$, then each irreducible component has a codimension distinct from the others, and the least of these codimensions is the codimension of the set of matrix pencils with rank at most r. This is $(n-r)(2m-r)$ if $m \geq n$ and $(m-r)(2n-r)$ otherwise.

Key words. Kronecker canonical form, matrix pencils, orbits, closures, irreducible components

AMS subject classifications. 15A21, 15A22, 65F15

DOI. 10.1137/060662538

1. Introduction.

The Kronecker canonical form (KCF) [6, Chapter XII] of matrix pencils may reflect important physical properties of the systems modelled by pencils, such as controllability [4, 10]. Very significant advances in the development of algorithms to compute the KCF have been seen in recent years (see [5] and the references therein). Despite this fact, computing the KCF of matrix pencils is an expensive and delicate task. Therefore, theoretical results that describe generic KCFs of some subsets of matrix pencils, i.e., the KCFs of almost all pencils in the subset, are interesting from an applied point of view.

The [illegible] of full rank $m \times n$ complex matrix pencils $A - \lambda B$ with $m \neq n$ was explicitly described in [3, Corollary 7.1] (see also [5, section 3.3]). For $n \times n$ singular matrix pencils, there are n possible generic KCFs, each of them corresponding to an orbit of strictly equivalent matrix pencils of codimension $n+1$. These Kronecker structures were explicitly described in [11, Theorem 1] (see also [3, Corollary 7.2] and [5, section 3.3]). In rigorous mathematical terms, one can say, in the language of algebraic geometry [11], that the set of $n \times n$ singular matrix pencils has exactly n irreducible components of codimension $n+1$, or, in the language introduced in [5], that the set of $n \times n$ singular matrix pencils is the union of the closures of n maximal orbits of strictly equivalent matrix pencils. Another relevant result in this context is that the set of $m \times n$ matrices with rank at most r is a manifold in $\mathbb{C}^{mn}$ of codimension $(m-r)(n-r)$ [3, Lemma 3.3]. However, as far as we know, no similar results exist for pencils with rank at most r. To develop these kinds of results is the purpose of this paper.

*Received by the editors June 9, 2006; accepted for publication (in revised form) by B. T. Kågström January 3, 2008; published electronically May 2, 2008. This research was partially supported by the Ministerio de Educación y Ciencia of Spain through grants BFM-2003-00223, MTM-2006-05361, MTM-2006-06671, and by the PRICIT Program of the Comunidad de Madrid through SIMUMAT Project (Ref. S-0505/ESP/0158).

http://www.siam.org/journals/simax/30-2/66253.html

†Departamento de Matemáticas, Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés, Spain (fteran@math.uc3m.es, dopico@math.uc3m.es).

We prove that there are exactly $r+1$ generic KCFs for $m \times n$ pencils with rank r. More precisely, for $r = 1, \ldots, \min\{m,n\} - 1$, we show that the set of matrix pencils with rank [illegible] r is the union of the closures of the orbits corresponding to these KCFs, and that these closures are maximal in the sense that they are not contained in the closure of any other orbit of pencils with rank at most r. In addition, the dimensions of these orbits are counted and their KCFs explicitly described. The generic KCFs of the pencils with rank r have no regular part, as it happens for the generic KCFs of full rank $m \times n$ pencils and of $n \times n$ singular pencils [3, Corollaries 7.1 and 7.2], and they have both right and left singular blocks. Each of the generic KCFs with rank r depends on the sum of the right minimal indices [6], which may take values $0, 1, \ldots, r$, or, equivalently, on the sum of the left minimal indices. It is important to note that the orbits corresponding to these $r+1$ generic KCFs have different dimensions in the case $m \neq n$. To prove these results, we use techniques introduced in [3, 5] (see also [1]). Finally, we present an additional result on the irreducibility of the closures of the orbits of the generic KCFs in the Zariski topology [11, section 1]. Our results include, as a particular case, the KCFs of generic $n \times n$ singular matrix pencils.

The [illegible] of the pencil $A - \lambda B$ is defined in [6, Chapter VI] as the order of its largest minor that is not equal to the zero polynomial in λ. This is also frequently called [illegible] [1, 5]. We will use the more classical name [illegible] throughout this note, because this concept corresponds to the usual rank of matrices if we consider a matrix pencil as a matrix with elements in the field of rational functions in λ.

The fact that the KCFs of generic pencils with rank r depend not only on the rank, but also on the sum of the right (or, equivalently, left) minimal indices is related to a recent result presented in [2]. In [2], the generic change of the KCF of a pencil under low rank perturbations is studied, and it is proved that this change depends on the rank of the perturbation, and also on the sum of its left and right minimal indices.

A different kind of generic singular matrix pencils is considered in [8]. The definition of [illegible] in [8, p. 250] can be useful to study the KCF of very sparse pencils, but it is different from the one that we use in this work. As explained above, our definition of generic KCFs means that the union of the closures of the corresponding orbits is the whole set of pencils with rank at most r, and that these orbits are maximal. This is also the concept used in [5, 11]. This implies, for instance, that most pencils with rank r and generic KCF have all of the entries different from zero.[1] However, in the sense of [8], generic $m \times n$ pencils with rank r do not have all of their entries different from zero [8, Lemma 3.1]; therefore, they are not generic in our sense. We have already remarked that we will describe explicitly the generic KCFs of pencils with rank r; see Theorem 3.2. This cannot be done for the generic pencils with rank r in the sense of [8], where only the sums of the minimal indices of the KCF can be implicitly determined [8, Theorems 7.2 and 7.3].

This paper is organized as follows: In section 2 some background is introduced. The main results—Theorems 3.2, 3.3, and 3.5—are presented in section 3.

2. Previous results. In this section we briefly summarize the results needed in this note. Simultaneously, the basic notation is introduced.

[1] Note that the set of pencils with rank at most r and all of the entries different from zero is open and dense in the set of pencils with rank at most r. To see this in $r = 1$, note that every pencil with rank at most one can be written as $p(\lambda)q(\lambda)^T$, where $p(\lambda)$ and $q(\lambda)$ are polynomial vectors, one of them of degree 0 and the other one of degree at most 1. Most pencils of this type have all of the entries different from zero.

2.1. Orbits and the Kronecker canonical form. We will use the same notation as in [5]. The orbit $\mathcal{O}(\mathcal{M})$ of an $m \times n$ matrix pencil $\mathcal{M}(\lambda) = A - \lambda B$ is the set of matrix pencils strictly equivalent to $\mathcal{M}(\lambda)$:

$$\mathcal{O}(\mathcal{M}) = \{P\mathcal{M}(\lambda)Q : \ P \in \mathbb{C}^{m\times m}, Q \in \mathbb{C}^{n\times n} \ , \ P, Q \text{ nonsingular}\}.$$

These orbits are manifolds in the vector space $\mathbb{C}^{2mn}$, and we will refer to the dimension of $\mathcal{O}(\mathcal{M})$ as the codimension in this space. We will denote by $\overline{\mathcal{O}}(\mathcal{M})$ the closure of this orbit.

The most significant element of the orbit $\mathcal{O}(\mathcal{M})$ is the Kronecker canonical form (KCF) (e.g., see [6]) of $\mathcal{M}(\lambda)$. The KCF is the direct sum of the right singular, left singular, and regular structures, consisting of L_k blocks of dimension $k \times (k+1)$ for the right singular structure and L_k^T blocks for the left singular structure. The regular structure consists of Jordan blocks $J_k(\mu)$ corresponding to eigenvalue μ, and N_k corresponding to the infinite eigenvalue. The KCF of $\mathcal{M}(\lambda)$ determines uniquely the orbit $\mathcal{O}(\mathcal{M})$, and, in particular, it fully determines the codimension of $\mathcal{O}(\mathcal{M})$ [3, Theorem 2.2].

2.2. Inclusion relationships between orbit closures. The relation $\geq$ in the set of sequences of nonnegative integers specifies that $(a_1, a_2, \ldots) \geq (b_1, b_2, \ldots)$ if $a_1 + \cdots + a_i \geq b_1 + \cdots + b_i$ for $i = 1, 2, \ldots$. We say that $(a_1, a_2, \ldots) > (b_1, b_2, \ldots)$ if $(a_1, a_2, \ldots) \geq (b_1, b_2, \ldots)$ and $(a_1, a_2, \ldots) \neq (b_1, b_2, \ldots)$ [5, section 2.1].

For every matrix pencil $\mathcal{M}(\lambda)$ with rank r, we consider the following three sequences defined in [5]:

$$\mathcal{R}(\mathcal{M}) + r = (r_0 + r, r_1 + r, r_2 + r, \ldots);$$

where r_i is the number of right singular blocks L_j in the KCF of $\mathcal{M}(\lambda)$ with $j \geq i$;

$$\mathcal{L}(\mathcal{M}) + r = (l_0 + r, l_1 + r, l_2 + r, \ldots),$$

where l_i is the number of left singular blocks L_j^T in the KCF of $\mathcal{M}(\lambda)$ with $j \geq i$; and, for every $\mu \in \mathbb{C} \cup \{\infty\}$,

$$\mathcal{J}_\mu(\mathcal{M}) + p = (w_1(\mu) + p, w_2(\mu) + p, \ldots),$$

where $w_i(\mu)$ is the number of Jordan blocks associated with the eigenvalue μ of dimension greater than or equal to i in the regular structure of the KCF of $\mathcal{M}(\lambda)$, and p is the number of right singular blocks in the KCF of $\mathcal{M}(\lambda)$. These sequences allow us to obtain inclusion relationships between the closures of the orbits of two different matrix pencils. This is presented in Theorem 2.1 below, obtained in [9], and later reformulated in [1] and [5]. We state the theorem as it appears in [5].

THEOREM 2.1 (see [5, Theorem 3.1]). Let $\mathcal{M}_1, \mathcal{M}_2$ be $m \times n$ matrix pencils with $p(\mathcal{M}_1)$ and $p(\mathcal{M}_2)$ the number of right singular blocks in their KCF's. Then $\overline{\mathcal{O}}(\mathcal{M}_1) \supseteq \overline{\mathcal{O}}(\mathcal{M}_2)$ if and only if

(i) $\mathcal{R}(\mathcal{M}_1) + \operatorname{rank}(\mathcal{M}_1) \geq \mathcal{R}(\mathcal{M}_2) + \operatorname{rank}(\mathcal{M}_2)$,
(ii) $\mathcal{L}(\mathcal{M}_1) + \operatorname{rank}(\mathcal{M}_1) \geq \mathcal{L}(\mathcal{M}_2) + \operatorname{rank}(\mathcal{M}_2)$,
(iii) $\mathcal{J}_\mu(\mathcal{M}_1) + p(\mathcal{M}_1) \leq \mathcal{J}_\mu(\mathcal{M}_2) + p(\mathcal{M}_2)$

for all $\mu \in \mathbb{C} \cup \{\infty\}$.

3. The set of singular pencils of rank at most r. If we restrict ourselves to the set of $m \times n$ matrix pencils with fixed rank equal to r, then the conditions in Theorem 2.1 simplify significantly. In this case, $\overline{\mathcal{O}}(\mathcal{M}_1) \supseteq \overline{\mathcal{O}}(\mathcal{M}_2)$ if and only if

the following conditions hold: (i) $\mathcal{R}(\mathcal{M}_1) \geq \mathcal{R}(\mathcal{M}_2)$, (ii) $\mathcal{L}(\mathcal{M}_1) \geq \mathcal{L}(\mathcal{M}_2)$, and (iii) $\mathcal{J}_\mu(\mathcal{M}_1) \leq \mathcal{J}_\mu(\mathcal{M}_2)$ for all $\mu \in \mathbb{C} \cup \{\infty\}$ (because $p(\mathcal{M}) = n - \operatorname{rank}(\mathcal{M})$).

We will make use of the following result whose proof is immediate.

LEMMA 3.1. [illegible] $\mathcal{T} = \operatorname{diag}(\mathcal{G}, \mathcal{H})$ [illegible] $\mathcal{G} \in \overline{\mathcal{O}}(\mathcal{M}_1)$ [illegible] $\mathcal{H} \in \overline{\mathcal{O}}(\mathcal{M}_2)$ [illegible] $\mathcal{M}_1$ [illegible] $\mathcal{M}_2$ [illegible] $\overline{\mathcal{O}}(\mathcal{T}) \subseteq \overline{\mathcal{O}}(\operatorname{diag}(\mathcal{M}_1, \mathcal{M}_2))$

Our main result characterizes the set of singular pencils with rank at most r through a set of maximal orbits of pencils with rank exactly r.

THEOREM 3.2. [illegible] r [illegible] $1 \leq r \leq \min\{m,n\}-1$ [illegible] $m \times n$ [illegible] r [illegible] $r+1$ [illegible]

$$\mathcal{K}_a(\lambda) = \operatorname{diag}(\underbrace{L_{\alpha+1}, \ldots, L_{\alpha+1}}_{s}, \underbrace{L_\alpha, \ldots, L_\alpha}_{n-r-s}, \underbrace{L_{\beta+1}^T, \ldots, L_{\beta+1}^T}_{t}, \underbrace{L_\beta^T, \ldots, L_\beta^T}_{m-r-t}) \tag{1}$$

[illegible] $a = 0, 1, \ldots, r$ [illegible] $\alpha = \lfloor a/(n-r) \rfloor$, $s = a \bmod (n-r)$, $\beta = \lfloor (r-a)/(m-r) \rfloor$ [illegible] $t = (r-a) \bmod (m-r)$ [illegible]

(i) [illegible] $m \times n$ [illegible] $\mathcal{M}(\lambda)$ [illegible] r [illegible] a [illegible] $\overline{\mathcal{O}}(\mathcal{K}_a) \supseteq \overline{\mathcal{O}}(\mathcal{M})$

(ii) $\overline{\mathcal{O}}(\mathcal{K}_a) \not\supseteq \overline{\mathcal{O}}(\mathcal{K}_{a'})$ [illegible] $a \neq a'$

(iii) [illegible] $m \times n$ [illegible] r [illegible] $\bigcup_{0 \leq a \leq r} \overline{\mathcal{O}}(\mathcal{K}_a)$

[illegible]. For each $a = 0, 1, \ldots, r$, let $\mathcal{D}_a$ be the set of block-diagonal matrix pencils in the form $\operatorname{diag}(\mathcal{G}, \mathcal{H})$, where $\mathcal{G}$ and $\mathcal{H}$ are, respectively, $a \times (a+n-r)$ and $(m-a) \times (r-a)$ matrix pencils. The generic KCFs of $\mathcal{G}$ and $\mathcal{H}$ are, respectively, $\operatorname{diag}(L_{\alpha+1}, \overset{s)}{\ldots}, L_{\alpha+1}, L_\alpha, \overset{n-r-s)}{\ldots}, L_\alpha)$ and $\operatorname{diag}(L_{\beta+1}^T, \overset{t)}{\ldots}, L_{\beta+1}^T, L_\beta^T, \overset{m-r-t)}{\ldots}, L_\beta^T)$ (where $\overset{k)}{\ldots}$ means that there is a series of exactly k equal terms), with α, β, s, and t as in the statement (see [3, Corollary 7.1]). Now, to prove the first part of the theorem, it remains to show only that any matrix pencil of rank at most r is strictly equivalent to a block-diagonal pencil in $\mathcal{D}_a$, for some $a = 0, 1, \ldots, r$, and apply Lemma 3.1.

Let $\mathcal{M}(\lambda)$ be a matrix pencil with rank $r' \leq r$ and KCF given by

$$\mathcal{K}_{\mathcal{M}}(\lambda) = \operatorname{diag}(L_{\alpha_1}, \ldots, L_{\alpha_{n-r'}}, L_{\beta_1}^T, \ldots, L_{\beta_{m-r'}}^T, J),$$

where J is the regular structure of the KCF. Then, since $r' \leq r$, we can consider

$$\mathcal{G} = \operatorname{diag}(L_{\alpha_1}, \ldots, L_{\alpha_{n-r}})$$

and $\mathcal{H}$ being the block-diagonal matrix pencil containing the remaining blocks in $\mathcal{K}_{\mathcal{M}}(\lambda)$. Notice that $\mathcal{G}$ is of size $a \times (a+n-r)$, with $a = \alpha_1 + \cdots + \alpha_{n-r}$. Then $\mathcal{M}(\lambda)$ is equivalent to $\operatorname{diag}(\mathcal{G}, \mathcal{H})$, and this last matrix pencil is in the class $\mathcal{D}_a$.

Now, we show that $\overline{\mathcal{O}}(\mathcal{K}_a) \not\supseteq \overline{\mathcal{O}}(\mathcal{K}_{a'})$ whenever $a \neq a'$. For this, it suffices to check that for distinct $a, a' \in \{0, 1, \ldots, r\}$ the simplified versions of the three conditions (i), (ii), and (iii) in Theorem 2.1 do not hold simultaneously. This fact is immediate, because $a > a'$ implies (with the same notation as in the statement)

$$\alpha > \alpha' \quad \text{or} \quad \alpha = \alpha' \text{ and } s > s',$$

which implies $\mathcal{R}(\mathcal{K}_a) > \mathcal{R}(\mathcal{K}_{a'})$, and also

$$\beta < \beta' \quad \text{or} \quad \beta = \beta' \text{ and } t < t',$$

which implies $\mathcal{L}(\mathcal{K}_a) < \mathcal{L}(\mathcal{K}_{a'})$.

Finally, notice that the third item in Theorem 3.2 is a direct consequence of the first item, $\operatorname{rank}(\mathcal{K}_a(\lambda)) = r$ for all a, and the fact that the set of pencils of rank at most r is closed. □

Theorem 3.2 is, in essence, a consequence of Corollary 7.1 in [3], though we have stated it using the concepts and terminology from [5]. Notice also that although the rank of the KCFs in (1) is exactly r, the closures of their orbits include the set of all pencils with rank smaller than or equal to r.

Next, we pay attention to the codimension of the orbits $\mathcal{O}(\mathcal{K}_a)$ of the generic KCFs of pencils with rank at most r. We will see that these codimensions are distinct if $m \neq n$. In this case, the codimension (dimension) of the set of matrix pencils with rank at most r is defined, according to [11], as the least (largest) of the codimensions (dimensions) of $\mathcal{O}(\mathcal{K}_a)$, for $a = 0, 1, \ldots, r$.

THEOREM 3.3. [illegible] r [illegible] $1 \leq r \leq \min\{m, n\} - 1$ [illegible] $\mathcal{K}_a(\lambda)$ [illegible] $a = 0, 1, \ldots, r$ [illegible] $r + 1$ [illegible] (1) [illegible]

1. [illegible] $\mathcal{O}(\mathcal{K}_a)$ [illegible] $(n - r)(2m - r) + a(m - n)$.
2. [illegible] $m \times n$ [illegible] r [illegible]
   (i) $(n - r)(2m - r)$ [illegible] $m \geq n$ [illegible]
   (ii) $(m - r)(2n - r)$ [illegible] $m \leq n$

[illegible]. The first item is a direct consequence of [3, Theorem 2.2]. The second item follows from computing $\min_a\{(n - r)(2m - r) + a(m - n)\}$. □

3.1. Irreducibility in Zariski topology. All of the topological ideas used so far refer to the usual topology in $\mathbb{C}^{2mn}$. The Zariski topology was used by Waterhouse to prove that the set of $n \times n$ singular matrix pencils with entries in an arbitrary infinite field has exactly n irreducible components, each of codimension $n + 1$ [11, Theorem 1]. In this subsection, we will prove that the closures $\overline{\mathcal{O}}(\mathcal{K}_a(\lambda))$ of the orbits of the KCFs appearing in (1) are irreducible in the Zariski topology. A clear and concise summary of Zariski topology appears in the introduction of [11]. Here we recall only the following ideas: (i) a subset of $\mathbb{C}^q$ is closed in the Zariski topology if it is the set of common zeros of some polynomials, (ii) Zariski-closed sets are closed in the usual sense but the opposite is not true, (iii) a subset of $\mathbb{C}^q$ is irreducible if it is not the union of two relatively closed proper subsets in the Zariski topology, and (iv) every Zariski-closed set is the finite union of maximal irreducible subsets called its irreducible components.

An important result in this context is that the closures of an orbit of strictly equivalent pencils are the same in both the Zariski and the usual topology of $\mathbb{C}^{2mn}$ [7]. Therefore, there is no ambiguity in using the symbol $\overline{\mathcal{O}}(\mathcal{K}_a)$ in this subsection because it refers to exactly the same set as in the rest of this paper. The main result in this section states that orbits of pencils are irreducible.

LEMMA 3.4. [illegible] $\overline{\mathcal{O}}(\mathcal{M})$ [illegible] $\mathcal{O}(\mathcal{M})$ [illegible] $m \times n$ [illegible] $\mathcal{M}(\lambda)$ [illegible]

[illegible]. Let us identify the set of matrix pencils of size $m \times n$ with $\mathbb{C}^{2mn}$, where the pencil $\mathcal{M}(\lambda) = A - \lambda B$ is identified with the pair (A, B). Let U be the set of pairs (P, Q) with $P \in \mathbb{C}^{m \times m}$ and $Q \in \mathbb{C}^{n \times n}$ nonsingular. This is a dense open set of $\mathbb{C}^{m^2+n^2}$ (that is, $\overline{U} = \mathbb{C}^{m^2+n^2}$). Given a matrix pencil (A, B), we can consider

the continuous (polynomial) mapping $\varphi_{\mathcal{M}}$ from $\mathbb{C}^{m^2+n^2}$ to $\mathbb{C}^{2mn}$ defined by sending (P,Q) to (PAQ, PBQ). We have $\mathcal{O}(\mathcal{M}) = \varphi_{\mathcal{M}}(U)$. By [11, section 1] we know that $\overline{\varphi_{\mathcal{M}}(\mathbb{C}^{m^2+n^2})}$ is an irreducible set. On the other hand, for every continuous mapping φ, we have $\varphi(\overline{W}) \subset \overline{\varphi(W)}$, where W is an arbitrary set, and this implies $\overline{\varphi(\overline{W})} = \overline{\varphi(W)}$. In the present case, we have

$$\overline{\varphi_{\mathcal{M}}(U)} = \overline{\varphi_{\mathcal{M}}(\mathbb{C}^{m^2+n^2})},$$

and this equals $\overline{\mathcal{O}}(\mathcal{M})$, which concludes the proof. □

With this lemma, Theorem 3.2 can be complemented as follows.

THEOREM 3.5. [illegible] $1 \leq r \leq \min\{m,n\} - 1$, [illegible] $\mathcal{K}_a(\lambda)$, [illegible] $a = 0, 1, \ldots, r$, [illegible] $r+1$ [illegible] (1) [illegible] $m \times n$ [illegible] r [illegible] $r+1$. [illegible] $\overline{\mathcal{O}}(\mathcal{K}_a)$, $a = 0, 1, \ldots, r$

This theorem includes [11, Theorem 1], mentioned at the beginning of this section, as a particular case.

Acknowledgments. The authors thank two anonymous referees for suggesting the use of block-diagonal pencils to prove Theorem 3.2. This idea has enhanced the presentation very much. The authors also thank the editor, Prof. Bo Kågström, for pointing out reference [8].

REFERENCES

[1] I. DE HOYOS, *Points of continuity of the Kronecker canonical form*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 278–300.

[2] F. DE TERÁN AND F. M. DOPICO, *Low rank perturbation of Kronecker structures without full rank*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 496–529.

[3] J. DEMMEL AND A. EDELMAN, *The dimension of matrices (matrix pencils) with given Jordan (Kronecker) canonical forms*, Linear Algebra Appl., 230 (1995), pp. 61–87.

[4] J. DEMMEL AND B. KÅGSTRÖM, *Accurate solutions of ill-posed problems in control theory*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 126–145.

[5] A. EDELMAN, E. ELMROTH, AND B. KÅGSTRÖM, *A geometric approach to perturbation theory of matrices and matrix pencils.* II. *A stratification-enhanced staircase algorithm*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 667–699.

[6] F. GANTMACHER, *The Theory of Matrices*, Vols. 1 and 2, Chelsea Publishing Co., New York, NY, 1959.

[7] D. HINRICHSEN AND J. O'HALLORAN, *Orbit closures of singular matrix pencils*, J. Pure Appl. Algebra, 81 (1992), pp. 117–137.

[8] S. IWATA AND R. SHIMIZU, *Combinatorial analysis of singular matrix pencils*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 245–259.

[9] A. POKRZYWA, *On perturbations and the equivalence orbit of a matrix pencil*, Linear Algebra Appl., 82 (1986), pp. 99–121.

[10] P. M. VAN DOOREN, *The generalized eigenstructure problem in linear system theory*, IEEE Trans. Automat. Control, 26 (1981), pp. 111–128.

[11] W. WATERHOUSE, *The codimension of singular matrix pairs*, Linear Algebra Appl., 57 (1984), pp. 227–245.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 497–508

ON INVERSES OF TRIDIAGONAL MATRICES ARISING FROM MARKOV CHAIN-RANDOM WALK I*

MOHAMED A. EL-SHEHAWEY†

Abstract. Explicit expressions for the entries of the inverse of a general diagonal matrix are derived via the decomposition approach. In particular, three-term recurrence relations with variable coefficients are given, and with their solutions closed form expressions for each entry of the fundamental matrix of a Markov chain-random walk (MC-RW) model are established. We also obtain explicit formulas for the absorption probabilities and the mean time to absorption as well as the mean number of steps before absorption of the MC-RW model in the presence of two semiabsorbing boundaries.

Key words. tridiagonal matrix, three-term recurrence relations, Markov chain-random walk, semiabsorbing barriers, time to absorption

AMS subject classifications. 60G40, 60J80

DOI. 10.1137/040618448

1. Introduction. There is an intimate relationship between Markov chain-random walk (MC-RW), tridiagonal matrices, and three-term recurrence relations. The latter two are known to be powerful tools for counting walks with various specifications, are frequently used to describe the motion of a particle in a one-dimensional chain. Tridiagonal matrices appear frequently in various branches of mathematics, modern physics, and engineering. Therefore this class of matrices is studied extensively and a great deal of theory is known about their inverses. For historical remarks and physical motivations, cf. [21, 20, 29, 6, 19, 25, 10, 1, 15]. Three-term recurrence relations lie at the heart of continued fractions, orthogonal polynomials, and birth-death processes, cf. [7, 4, 30, 16, 17, 18, 3, 2]. The MC-RW is a venerable model, finding its applications in many areas including biology, chemistry, and physics, but it is also a useful tool in randomized algorithms in computer science. Despite its long history, novel aspects continue to surface. For comprehensive treatments of MC-RW and its applications, cf. [5, 11, 28, 13, 25, 8, 12, 23, 9]. The MC-RW is a stochastic process $X = \{X_k : k \geq 0\}$ characterized by the transition matrix, $P = (p_{ij})_{ij}$, given via the entries of the three vectors $q = (q_0, q_1, \ldots, q_n)$, $r = (r_0, r_1, \ldots, r_n)$, $p = (p_0, p_1, \ldots, p_n)$ as $P = P(q, r, p)$ with

$$P = \begin{pmatrix} r_0 & p_0 & 0 & & \cdots & 0 \\ q_1 & r_1 & p_1 & 0 & & \vdots \\ 0 & q_2 & r_2 & p_2 & \ddots & \\ & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & q_{n-1} & r_{n-1} & p_{n-1} \\ 0 & \cdots & & 0 & q_n & r_n \end{pmatrix}. \tag{1.1}$$

*Received by the editors November 8, 2004; accepted for publication (in revised form) by A. Frommer January 9, 2008; published electronically May 9, 2008.
http://www.siam.org/journals/simax/30-2/61844.html
†Department of Mathematics, Damiettta Faculty of Science, El-Mansoura University, New Damietta, Egypt (el_shehawy@mans.edu.eg).

Here, $p_{ij} = \Pr(X_{k+1} = j \mid X_k = i)$ is the one-step transition probability. The definition of an MC-RW implies that $r_j + p_j + q_j = 1$ and $r_j,\ p_j,\ \ q_j \geq 0$. The boundary probabilities p_0, r_0, r_n, and q_n reflect boundary conditions of the model under investigation. We allow $q_0 = 1 - p_0 - r_0 \geq 0$, $p_n = 1 - q_n - r_n \geq 0$. If $q_0 = 0$ and $p_n = 0$, then the matrix P is stochastic and the MC-RW is said to have reflecting barriers at 0 and n. If $q_0 > 0$ (or $p_n > 0$), then the MC-RW has an (ignored) absorbing state at -1 (or at $n+1$), which can be reaching through state 0 (or n) only. In many stochastic models, in particular Markov chains in discrete or continuous time and Markov renewal processes, a Markov chain arises either directly or indirectly through some form of embedding. The analysis of many problems of interest associated with these models, e.g., the moments of first passage time distributions and the moments of occupation time random variables, often concerns the inverse of the matrix $I - P$. In the present paper we wish to obtain closed form expressions for each entry of the fundamental matrix of the MC-RW model which are not readily available in the literature on either matrix theory or probability theory. This leads to explicit expressions for the aforementioned problems. Many interesting particular cases can be derived from this study through an appropriate choice of the boundary probabilities p_0, q_n, r_0, and r_n. We shall restrict our attention to MC-RW in the presence of two semiabsorbing boundaries, $p_0 = \rho$, $q_n = \omega$, and $r_0 = r_n = 0$. Physically, this corresponds to the situation where upon reaching the barrier 0 (or n) the particle is either lost from the system with probability $1 - \rho$ (or $1 - \omega$) or turned back to the system with probability ρ (or ω) and reduces to the classical problem of random walk. This leads to a list of particular cases: (i) MC-RW in the presence of two symmetric semiabsorbing barriers at 0 and n, for $\rho = \omega$, $0 < \rho < 1$, and $0 < \omega < 1$. (ii) MC-RW in the presence of two asymmetric partially reflecting barriers at 0 and n, for $\rho \neq \omega$, $0 < \rho < 1$ and $0 < \omega < 1$. (iii) MC-RW in the presence of two perfectly absorbing barriers at 0 and n, for $\rho = \omega = 0$. (iv) MC-RW in the presence of two perfectly reflecting barriers at 0 and n, for $\rho = \omega = 1$. (v) MC-RW in the presence of two different barriers, one of which at 0 (or at n) is perfectly absorbing and the other at n (or at 0) is semireflecting, for $\rho = 0$, $0 < \omega < 1$ (or $0 < \rho < 1$, $\omega = 0$), respectively. (vi) MC-RW in the presence of two different barriers, one of which at 0 (or at n) is perfectly absorbing and the other at n (or at 0) is perfectly reflecting, for $\rho = 0, \omega = 1$ (or $\rho = 1, \omega = 0$), respectively. (vii) MC-RW having a single (partially or perfectly) absorbing barrier at the origin, by taking the limit when $n \to \infty$.

A computational approach to finding the inverse of the general tridiagonal matrix P in (1.1) is given in a recent paper by Mallik [19]. To do this he imposed the restrictive conditions $p_0, p_1, \ldots, p_{n-1} \neq 0$ and $q_1, q_2, \ldots, q_n \neq 0$. In section 2 of the present paper, the elements of the inverse of the tridiagonal matrix P in (1.1), without imposing any restrictive conditions, via the Cholesky decompositional approach, are expressed in terms of the determinants of tridiagonal submatrices of P in (1.1). These determinants are solutions of a linear second-order difference equation with variable coefficients. In section 3, difference equations with variable coefficients are resolved explicitly. With these simple solutions, new results are obtained in sections 4 and 5; explicit expressions for the elements of the fundamental matrix of the model under investigation, with some interesting particular cases, are presented in section 4. In section 5, the results are applied to the MC-RW problem, in the presence of two semiabsorbing boundaries.

2. Inverse of the tridiagonal matrix P. Consider a general $(n+1)\times(n+1)$ tridiagonal matrix P of the form (1.1). For suitable (unique) diagonal matrices

$$D_1 = \mathrm{diag}(\alpha_0, \alpha_1, \ldots, \alpha_n),\ D_2 = \mathrm{diag}(\gamma_0, \gamma_1, \ldots, \gamma_n), \tag{2.1}$$

the matrix P can be decomposed into the following two factorizations:

$$P = U_1 D_1^{-1} L_1 \text{ and } P = L_2 D_2^{-1} U_2, \tag{2.2}$$

where

$$\begin{array}{l} U_1 = P\,(0, \alpha, p)\,, L_1 = P\,(q, \alpha, 0)\,, \\ U_2 = P\,(q, \gamma, 0)\,, L_2 = P\,(0, \gamma, p)\,, \end{array} \tag{2.3}$$

and D_k^{-1} is the inverse of D_k, $k = 1, 2$. Equating the entries on either side of (2.2), respectively, gives

$$\alpha_i = \begin{cases} r_i - \frac{p_i\, q_{i+1}}{\alpha_{i+1}} & \text{for } i = 0, 1, \ldots, n-1, \\ r_n & \text{for } i = n, \end{cases} \tag{2.4}$$

$$\gamma_i = \begin{cases} r_0 & \text{for } i = 0, \\ r_i - \frac{p_{i-1}\, q_i}{\gamma_{i-1}} & \text{for } i = 1, 2, \ldots, n. \end{cases} \tag{2.5}$$

The elements of the inverse P^{-1}can be computed according to the following theorem (cf. [2, 6, 14, 19, 29]). We use the convention that empty products and terms that have indices greater than n must be taken equal to 1.

THEOREM 2.1. [illegible] P [illegible] (1.1) [illegible] $Q = P^{-1} = (q_{ij})_{i,j}$ [illegible]

$$q_{ij} = \begin{cases} (-1)^{j-i} \left(\prod_{k=i}^{j-1} p_k \right) \frac{\alpha_{j+1}\alpha_{j+2}\cdots\alpha_n}{\gamma_i\gamma_{i+1}\cdots\gamma_n} & \text{[illegible]} \quad i \le j, \\ (-1)^{i-j} \left(\prod_{k=j+1}^{i} q_k \right) \left[\frac{\alpha_{i+1}\alpha_{i+2}\cdots\alpha_n}{\gamma_j\gamma_{j+1}\cdots\gamma_n} \right] & \text{[illegible]} \quad i \ge j. \end{cases} \tag{2.6}$$

[illegible] (2.5) [illegible] (2.4) [illegible]

$$\gamma_i = \frac{\Delta_i}{\Delta_{i-1}} \text{ [illegible]} \quad i = 1, 2, \ldots, n \qquad \text{[illegible]} \quad \Delta_{-1} = 1, \Delta_0 = r_0,$$

$$\alpha_i = \frac{\nabla_i}{\nabla_{i+1}} \text{ [illegible]} \quad i = 0, 1, 2, \ldots, n-1 \quad \text{[illegible]} \quad \nabla_{n+1} = 1, \nabla_n = r_n,$$

[illegible]

$$\Delta_i = r_i\Delta_{i-1} - p_{i-1}q_i\Delta_{i-2} \text{ [illegible]} \quad i = 1, 2, \cdots, n \tag{2.7}$$

[illegible] $\Delta_{-1} = 1$ [illegible] $\Delta_0 = r_0$ [illegible]

$$\nabla_i = r_i\nabla_{i+1} - p_iq_{i+1}\nabla_{i+2} \text{ [illegible]} \quad i = 0, 1, 2, \cdots, n-1 \tag{2.8}$$

[illegible] $\nabla_{n+1} = 1$ [illegible] $\nabla_n = r_n$

Rewriting formula (2.6), in terms of Δ_i and ∇_i, we get the following theorem.

THEOREM 2.2. [illegible] P [illegible] (1.1) [illegible] $Q = P^{-1} = (q_{ij})_{i,j}$ [illegible]

$$q_{ij} = \frac{1}{\Delta_n} \begin{cases} (-1)^{j-i} \left(\prod_{k=i}^{j-1} p_k \right) \Delta_{i-1} \nabla_{j+1} & \text{[illegible]} \quad i \le j, \\ (-1)^{i-j} \left(\prod_{k=j+1}^{i} q_k \right) \Delta_{j-1} \nabla_{i+1} & \text{[illegible]} \quad i \ge j, \end{cases} \tag{2.9}$$

[illegible] Δ_i [illegible] ∇_i [illegible] (2.7) [illegible] (2.8) [illegible]
[illegible] Δ_i [illegible] ∇_i [illegible]

$$\Delta_i = \det P\left((q_0, \dots, q_i), (r_0, \dots, r_i), (p_0, \dots, p_i)\right),$$
$$\nabla_i = \det P\left((q_i, \dots, q_n), (r_i, \dots, r_n), (p_i, \dots, p_n)\right).$$

[illegible] (2.7) [illegible] (2.8) [illegible]

$$\begin{aligned} \det P &= \alpha_0 \alpha_1 \cdots \alpha_{n-1} \alpha_n = \nabla_0 \\ &= \gamma_0 \gamma_1 \cdots \gamma_{n-1} \gamma_n = \Delta_n, \end{aligned}$$

[illegible] P [illegible] (2.7) [illegible] (2.8)

3. Solution of three-term recurrence relations. Difference equations with variable coefficients have been studied in many papers (cf. [2, 3, 16, 17, 18, 19, 24, 30]). In [16, 17, 18, 19], Mallik introduced the definition of a set $S_q(l+1, m)$ as follows: Let $\mathbb{N}$ denote the set of natural numbers. For $q, L, U \in \mathbb{N}$, $S_q(L, U)$ is the set of all q-tuples with elements from $\{L, L+1, \dots, U\}$ arranged in ascending order so that no consecutive elements are present. Mallik then proved the following result (cf. formulas (18) and (63a)–(63b) in [19].

PROPOSITION 3.1. [illegible]

$$E_m(l) = -\frac{r_m}{p_m} E_{m-1}(l) - \frac{q_{m-1}}{p_m} E_{m-2}(l), \quad 1 \le m \le n, 1 \le l \le m \tag{3.1}$$

[illegible]

$$E_m(l) := \begin{cases} \dfrac{(-1)^{m-l+1}}{\prod_{j=l}^{m} p_j} \left(\prod_{j=l}^{m} r_j + \sum_{q=1}^{\lfloor \frac{m-l+1}{2} \rfloor} \sum_{(k_1, \dots, k_q) \in S_q(l+1, m)} \sigma_{l,m}(k_1, \dots, k_q) \right) \\ \qquad \text{[illegible]} \quad l = 1, \dots, m-1; m = 2, \dots, n, \\ -\frac{r_m}{p_m} \quad \text{[illegible]} \quad l = m; \ m = 1, \dots, n, \\ 1 \quad \text{[illegible]} \quad l = m+1; \ m = 0, \dots, n, \\ 0 \quad \text{[illegible]} \end{cases} \tag{3.2}$$

[illegible]

$$E_{-1}(1) = 0, \quad E_m(m+1) = 1 \text{ [illegible]} \quad 0 \le m \le n, \ E_m(m) = -\frac{r_m}{p_m} \text{ [illegible]} \quad 1 \le m \le n,$$

$$\sigma_{l,m}(k_1,\dots,k_q) := (-1)^q \left(\prod_{j=l}^{m} r_j\right) \prod_{i=1}^{q} \frac{p_{k_i-1}q_{k_i-1}}{r_{k_i-1}r_{k_i}}$$

$E_m(l)$ (3.2)

LEMMA 3.1. (2.7) (2.8)

$$\Delta_i = r_0 \prod_{j=1}^{i} \gamma_j \quad \nabla_i = r_n \prod_{j=i}^{n-1} \alpha_j, \tag{3.3}$$

γ_j α_j (2.5) (2.4) . The proof is easily obtained. □

With the assumptions that $r_m + p_m + q_m = 1$, $r_0 + p_0 \geq 0$, and $r_n + q_n \geq 0$, we obtain the following corollary.

COROLLARY 3.1.

$$\begin{cases} Y_m = (1-r_m)Y_{m-1} - p_{m-1}q_m Y_{m-2}, & m \geq 1, \\ Y_0 = 1 - r_0 \quad Y_{-1} = 1, \end{cases} \tag{3.4}$$

$$\begin{cases} Y_m = (1-r_m)Y_{m+1} - p_m q_{m+1} Y_{m+2}, & m < n, \\ Y_n = 1 - r_n \quad Y_{n+1} = 1, \end{cases} \tag{3.5}$$

$$Y_m = (1-r_0-p_0)\sum_{u=0}^{m}\left(\prod_{j=1}^{m-u} q_j\right)\left(\prod_{j=m-u+1}^{m} p_j\right) + \left(\prod_{j=0}^{m} p_j\right), \quad m \geq 1, \tag{3.6}$$

$$Y_m = (1-r_n-q_n)\sum_{u=0}^{n-m}\left(\prod_{j=m}^{m+u-1} q_j\right)\left(\prod_{j=m+u}^{n-1} p_j\right) + \left(\prod_{j=m}^{n} q_j\right), \quad m < n \tag{3.7}$$

COROLLARY 3.2.

$$\begin{cases} Y_m = (1-r_m)Y_{m-1} - p_{m-1}q_m Y_{m-2}, & m \geq 2, \\ Y_1 = 1 - r_1 \quad Y_0 = 1, \end{cases} \tag{3.8}$$

$$\begin{cases} Y_m = (1-r_m)Y_{m+1} - p_m q_{m+1} Y_{m+2}, & m < n-1, \\ Y_{n-1} = 1 - r_{n-1} \quad Y_n = 1, \end{cases} \tag{3.9}$$

$$Y_m = \sum_{u=0}^{m}\left(\prod_{j=1}^{m-u} q_j\right)\left(\prod_{j=m-u+1}^{m} p_j\right), \quad m \geq 1, \tag{3.10}$$

$$Y_m = \sum_{u=0}^{n-m}\left(\prod_{j=m}^{m+u-1} q_j\right)\left(\prod_{j=m+u}^{n-1} p_j\right), \quad m < n \tag{3.11}$$

COROLLARY 3.3. $r + p + q = 1$

$$\begin{cases} Y_m = (1-r)Y_{m-1} - pqY_{m-2}, & m = 2,3,\dots,n-1, \\ Y_1 = 1 - r - \rho q \quad Y_0 = 1, \end{cases} \tag{3.12}$$

$$\begin{cases} Y_m = (1-r)Y_{m+1} - pqY_{m+2}, & m = n-2, n-1, \dots, 2, 1, \\ Y_{n-1} = 1 - r - p\omega \quad Y_n = 1, \end{cases} \tag{3.13}$$

$$Y_m = \frac{1}{p-q}\left[\left(1-\frac{q\rho}{p}\right)p^{m+1} - (1-\rho)q^{m+1}\right], \tag{3.14}$$

$$Y_{n-m} = \frac{1}{p-q}\left[(1-\omega)p^{m+1} - \left(1-\frac{p\omega}{q}\right)q^{m+1}\right]. \tag{3.15}$$

4. Closed form expressions for the fundamental matrix. Consider a finite-state MC-RW whose states are numbered so that $T = \{0, 1, \ldots, n\}$ denotes the set of transient states. Note that since P, as in (1.1), specifies only the transition probabilities from transient-to-transient states, some of its row sums are less than 1. Let $N_{ij}(n)$ be the number of visits to state j by time n given that $X_0 = i$; then $\{N_{ij}(n), n \geq 1\}$ is a delayed renewal process. A renewal occurs whenever the chain enters state j. For transient states i and j, let $\mathbf{m}_{ij} = E[N_{ij}(\infty)]$ be the expected total number of time periods spent in state j, starting in i; then

$$\mathbf{m}_{ij} = \delta(i,j) + \sum_{k=0}^{n} p_{ik}\mathbf{m}_{kj},$$

where $\delta(i,j)$ is the Kronecker delta. This equality holds from the fact that $\mathbf{m}_{ij} = 0$ when k is a recurrent state.

Let $M = (\mathbf{m}_{ij})_{i,j}$; then

$$M = (I-P)^{-1}.$$

Therefore, the (i,j)th element of the fundamental matrix M, $\mathbf{m}_{ij}$ is the mean number of visits to transient state j from state i before the chain enters any absorbing state. For $i \in T, j \in T$ the probability of ever visiting state j, starting in state i, is given by

$$f_{ij} = \frac{\mathbf{m}_{ij}}{\mathbf{m}_{jj}}.$$

The next theorem is immediate from Theorem 2.1 and Corollary 3.1.

THEOREM 4.1. $M = (\mathbf{m}_{ij})_{i,j}$ $\mathbf{m}_{ij}$ $i,j = 0,1,2,\ldots,n$, $0 \leq r_0, r_n < 1$, $0 \leq r_0 + p_0 \leq 1$, $0 \leq r_n + q_n \leq 1$,

$$\mathbf{m}_{ij} = \frac{1}{\Delta_n}\begin{cases} \Delta_1 & i = j = 0, \\ p_0\Delta_2 & i = 0, j = 1, \\ \left(\prod_{k=i}^{j-1} p_k\right)\Delta_{i-1}\nabla_{j+1} & 1 \leq i \leq j \leq n-1, \\ \left(\prod_{k=j+1}^{i} q_k\right)\Delta_{j-1}\nabla_{i+1} & 1 \leq j < i \leq n-1, \\ q_n\nabla_{n-2} & i = n, j = n-1, \\ \nabla_{n-1} & i = j = n, \end{cases} \tag{4.1}$$

$$\nabla_j = (1-r_n-q_n)\sum_{u=0}^{n-j}\left(\prod_{k=j}^{j+u-1} q_k\right)\left(\prod_{k=j+u}^{n-1} p_k\right) + \left(\prod_{k=j}^{n} q_k\right), \tag{4.2}$$

$$\Delta_i = (1-r_0-p_0)\sum_{u=0}^{i}\left(\prod_{k=1}^{i-u} q_k\right)\left(\prod_{k=i-u+1}^{i} p_k\right) + \left(\prod_{k=0}^{i} p_k\right), \tag{4.3}$$

$$\Delta_n = \nabla_0 = (1 - r_0 - p_0)\left(\prod_{k=1}^{n} q_k\right) + (1 - r_n - q_n)\left(\prod_{k=0}^{n-1} p_k\right)$$
$$+(1 - r_0 - p_0)(1 - r_n - q_n)\sum_{u=0}^{n-1}\left(\prod_{k=1}^{n-u-1} q_k\right)\left(\prod_{k=n-u}^{n-1} p_k\right) \tag{4.4}$$

$$P_0 = P\left((q_1, \ldots, q_n), (r_1, \ldots, r_n), (p_1, \ldots, p_n)\right),$$
$$P_n = P\left((q_0, \ldots, q_{n-1}), (r_0, \ldots, r_{n-1}), (p_0, \ldots, p_{n-1})\right), \text{and}$$
$$P_{0,n} = P\left((q_1, \ldots, q_{n-1}), (r_1, \ldots, r_{n-1}), (p_1, \ldots, p_{n-1})\right)$$

2.2 3.1–3.3 $M_0 = (I - P_0)^{-1}$, $M_n = (I - P_n)^{-1}$ $M_{0,n} = (I - P_{0,n})^{-1}$.

COROLLARY 4.1. $M_0 = (m_{ij})_{i,j}$

$$m_{ij} = \frac{1}{\Delta_n}\begin{cases} \left(\prod_{k=i}^{j-1} p_k\right)\Delta_{i-1}\nabla_{j+1} & 1 \le i \le j \le n-1, \\ \left(\prod_{k=j+1}^{i} q_k\right)\Delta_{j-1}\nabla_{i+1} & 1 \le j < i \le n-1, \\ q_n\nabla_{n-2} & i = n, j = n-1, \\ \nabla_{n-1} & i = j = n, \end{cases} \tag{4.5}$$

$$\nabla_j = (1 - r_n - q_n)\sum_{u=0}^{n-j}\left(\prod_{k=j}^{j+u-1} q_k\right)\left(\prod_{k=j+u}^{n-1} p_k\right) + \left(\prod_{k=j}^{n} q_k\right) \tag{4.6}$$

$$\Delta_i = \sum_{u=0}^{i}\left(\prod_{k=1}^{i-u} q_k\right)\left(\prod_{k=i-u+1}^{i} p_k\right) \tag{4.7}$$

$$\Delta_n = \left(\prod_{k=1}^{n} q_k\right) + (1 - r_n - q_n)\sum_{u=0}^{n-1}\left(\prod_{k=1}^{n-u-1} q_k\right)\left(\prod_{k=n-u}^{n-1} p_k\right) \tag{4.8}$$

COROLLARY 4.2. $M_n = (m_{ij})_{i,j}$

$$m_{ij} = \frac{1}{\Delta_n}\begin{cases} \Delta_1 & i = j = 0, \\ p_0\Delta_2 & i = 0, j = 1, \\ \left(\prod_{k=i}^{j-1} p_k\right)\Delta_{i-1}\nabla_{j+1} & 1 \le i \le j \le n-1, \\ \left(\prod_{k=j+1}^{i} q_k\right)\Delta_{j-1}\nabla_{i+1} & 1 \le j < i \le n-1, \end{cases} \tag{4.9}$$

$$\nabla_j = \sum_{u=0}^{n-j}\left(\prod_{k=j}^{j+u-1} q_k\right)\left(\prod_{k=j+u}^{n-1} p_k\right) \tag{4.10}$$

$$\Delta_i = (1 - r_0 - p_0)\sum_{u=0}^{i}\left(\prod_{k=1}^{i-u} q_k\right)\left(\prod_{k=i-u+1}^{i} p_k\right) + \left(\prod_{k=0}^{i} p_k\right) \tag{4.11}$$

$$\Delta_n = \left(\prod_{k=0}^{n-1} p_k\right) + (1 - r_0 - p_0) \sum_{u=0}^{n-1} \left(\prod_{k=1}^{n-u-1} q_k\right) \left(\prod_{k=n-u}^{n-1} p_k\right) \tag{4.12}$$

COROLLARY 4.3. $M_{0,n} = (m_{ij})_{i,j}$

$$m_{ij} = \frac{1}{\Delta_n} \begin{cases} \left(\prod_{k=i}^{j-1} p_k\right) \Delta_{i-1} \nabla_{j+1} & 1 \le i \le j \le n-1, \\ \left(\prod_{k=j+1}^{i} q_k\right) \Delta_{j-1} \nabla_{i+1} & 1 \le j < i \le n-1, \end{cases} \tag{4.13}$$

$$\nabla_j = \sum_{u=0}^{n-j} \left(\prod_{k=j}^{j+u-1} q_k\right) \left(\prod_{k=j+u}^{n-1} p_k\right), \tag{4.14}$$

$$\Delta_i = \sum_{u=0}^{i} \left(\prod_{k=1}^{i-u} q_k\right) \left(\prod_{k=i-u+1}^{i} p_k\right), \tag{4.15}$$

$$\Delta_n = \sum_{u=0}^{n-1} \left(\prod_{k=1}^{n-u-1} q_k\right) \left(\prod_{k=n-u}^{n-1} p_k\right) \tag{4.16}$$

(4.13), (4.16) [27] $q_i = \beta$, $p_i = \alpha, i = 1, 2, \ldots, n-1$ 4.1 $M_1 = (I - P)^{-1}$.

COROLLARY 4.4. $M_1 = (h_{ij})_{i,j}$

$$h_{ij} = \frac{1}{\Sigma} \begin{cases} (1 - r_0) a_n - \beta p_0 a_{n-1} & i = j = 0, \\ p_0 \left[(1 - r_0) a_{n-1} - \beta p_0 a_{n-2}\right] & i = 0, j = 1, \\ \alpha^{j-i} \left[(1 - r_0) a_i - \beta p_0 a_{i-1}\right] \left[(1 - r_n) a_{n-j} - \alpha q_n a_{n-j-1}\right] & 1 \le i \le j, \\ \beta^{i-j} \left[(1 - r_0) a_j - \beta p_0 a_{j-1}\right] \left[(1 - r_n) a_{n-i} - \alpha q_n a_{n-i-1}\right] & j \le i \le n-1, \\ q_n \left[(1 - r_n) a_{n-1} - \alpha q_n a_{n-2}\right] & i = n, j = n-1, \\ (1 - r_n) a_n - \alpha q_n a_{n-1} & i = j = n, \end{cases} \tag{4.17}$$

$$\begin{aligned} \Sigma = {} & (1 - r_0)(1 - r_n) a_n + \alpha\beta p_0 q_n a_{n-2} \\ & - \left[(1 - r_0)\alpha q_n + (1 - r_n)\beta p_0\right] a_{n-1}, \end{aligned} \tag{4.18}$$

$$a_i = \begin{cases} \frac{\alpha^i - \beta^i}{\alpha - \beta} & \alpha \ne \beta, \\ i\alpha^{i-1} & \alpha = \beta. \end{cases}$$

5. MC-RW in the presence of two semiabsorbing boundaries. In this section, we derive the absorption probabilities of an asymmetric one-dimensional random walk on the integers $\{0, 1, \ldots, n\}$ from the mean of the occupation totals in the presence of partially absorbing barriers at 0 and n. When the particle hits the barrier 0 (or n) it is either annihilated with probability $1 - \rho$ (or $1 - \omega$) or reflected back to the system with probability ρ (or ω), $0 \le \rho, \omega \le 1$. Physically, this corresponds to the situation when, reaching the barrier 0 (or n), the particle is either lost from the system with probability $1 - \rho$ (or $1 - \omega$) or tuned back to the system with probability ρ (or ω) and reduces to the classical problem of random walk. Assume that

X_k, $k = 1, 2, \ldots, n-1$ denotes a random variable associated with the kth step of the particle such that $X_k = 0$ if it stays put and is equal to 1 or -1 if it moves one unit to the right or to the left with respective probabilities γ, α and β; $\beta + \gamma + \alpha = 1$. Thus the present problem may be governed by the one-step transition probability matrix $P = P\left((1-\rho, \beta, \ldots, \beta, \omega), (0, \gamma, \ldots, \gamma, 0), (\rho, \alpha, \ldots, \alpha, 1-\omega)\right)$. Explicit expressions are investigated for the mean h_{ij} of the occupation totals, from which the absorption probabilities are determined as well as the mean time to absorption and the mean number of steps taken before absorption. Many interesting particular cases can be derived from the results through an appropriate choice of the reflection probabilities ρ and ω; see the introduction. After some simplifications, Corollary 4.3 gives

$$h_{0j} = \frac{1}{(1-\frac{\rho\beta}{\alpha})\Sigma} \begin{cases} \left(\frac{\alpha}{\beta}\right)^n - \frac{\beta-\omega\alpha}{\beta(1-\omega)} & \text{for } j = 0; \alpha \neq \beta, \\ \frac{\rho}{\alpha}\left[\left(\frac{\alpha}{\beta}\right)^{n-j} - \frac{\beta-\omega\alpha}{\beta(1-\omega)}\right]\left(\frac{\alpha}{\beta}\right)^j & \text{for } j = 1, 2, \ldots, n; \alpha \neq \beta, \end{cases}$$

$$= \frac{1}{(1-\rho)\Sigma_0} \begin{cases} n + \frac{\omega}{1-\omega} & \text{for } j = 0; \alpha = \beta, \\ \frac{\rho}{\alpha}\left[(n-j) + \frac{\omega}{1-\omega}\right] & \text{for } j = 1, 2, \ldots, n; \alpha = \beta, \end{cases} \tag{5.1}$$

$$h_{ij} = \frac{1}{(\alpha-\beta)\Sigma} \begin{cases} \left(\left(\frac{\alpha}{\beta}\right)^j - \frac{\alpha(1-\rho)}{\alpha-\beta\rho}\right)\left(\left(\frac{\alpha}{\beta}\right)^{n-i} - \frac{\beta-\omega\alpha}{\beta(1-\omega)}\right) & \\ \qquad \text{for } j \leq i; \alpha \neq \beta, & \\ \left(\left(\frac{\alpha}{\beta}\right)^i - \frac{\alpha(1-\rho)}{\alpha-\beta\rho}\right)\left(\left(\frac{\alpha}{\beta}\right)^{n-i} - \frac{\beta-\omega\alpha}{\beta(1-\omega)}\left(\frac{\alpha}{\beta}\right)^{j-i}\right) & \\ \qquad \text{for } j \geq i; \alpha \neq \beta & \end{cases}$$

$$= \frac{1}{\beta\Sigma_0} \begin{cases} \left(j + \frac{\rho}{1-\rho}\right)\left((n-i) + \frac{\omega}{1-\omega}\right) & \text{for } j \leq i; \alpha = \beta, \\ \left(i + \frac{\rho}{1-\rho}\right)\left((n-j) + \frac{\omega}{1-\omega}\right) & \text{for } j \geq i; \alpha = \beta, \end{cases} \tag{5.2}$$

$$h_{nj} = \frac{1}{(1-\omega)\Sigma} \begin{cases} \frac{\omega}{\beta}\left(\left(\frac{\alpha}{\beta}\right)^j - \frac{\alpha(1-\rho)}{\alpha-\beta\rho}\right) & \text{for } j = 0, 1, 2, \ldots, n-1; \alpha \neq \beta, \\ \left(\frac{\alpha}{\beta}\right)^n - \frac{\alpha(1-\rho)}{\alpha-\beta\rho} & \text{for } j = n; \alpha \neq \beta, \end{cases}$$

$$= \frac{1}{(1-\omega)\Sigma_0} \begin{cases} \frac{\omega}{\beta}\left(j + \frac{\rho}{1-\rho}\right) & \text{for } j = 0, 1, 2, \ldots, n-1; \alpha = \beta, \\ n + \frac{\rho}{1-\rho} & \text{for } j = n; \alpha = \beta, \end{cases} \tag{5.3}$$

where

$$\Sigma = \left(\frac{\alpha}{\beta}\right)^n - \frac{\alpha(1-\rho)(\beta-\alpha\omega)}{\beta(\alpha-\rho\beta)(1-\omega)}, \; \Sigma_0 = n + \frac{\rho}{1-\rho} + \frac{\omega}{1-\omega}. \tag{5.4}$$

Formulae (5.1)–(5.4) agree with the well-known result for MC-RW between two perfectly absorbing barriers, $\rho = \omega = 0$ (cf. [13, p. 100] in the case $\gamma = 0$).

Let $q(0 \mid i)$ and $q(n \mid i)$ denote the absorption probabilities at the boundary states 0 and n, respectively, given that i was the initial state. Therefore

$$q(0 \mid 0) = (1-\rho) \begin{cases} 1 + \frac{\rho\alpha}{(\alpha-\beta\rho)\Sigma}\left(\left(\frac{\alpha}{\beta}\right)^{n-1} - \frac{\beta-\alpha\omega}{\beta(1-\omega)}\right) & \text{for } \alpha \neq \beta, \\ 1 + \frac{\rho}{(1-\rho)\Sigma_0}\left(n - \frac{1-2\omega}{1-\omega}\right) & \text{for } \alpha = \beta, \end{cases} \tag{5.5}$$

$$(5.6)\quad q(n \mid n) = (1-\omega)\begin{cases} 1 + \frac{\omega\alpha}{\beta(1-\omega)\Sigma}\left(\left(\frac{\alpha}{\beta}\right)^{n-1} - \frac{\alpha(1-\rho)}{\alpha-\rho\beta}\right) & \text{for } \alpha \neq \beta, \\ 1 + \frac{\omega}{(1-\omega)\Sigma_0}\left(n - \frac{1-2\rho}{1-\rho}\right) & \text{for } \alpha = \beta, \end{cases}$$

$$(5.7)\quad q(0 \mid i) = \frac{1}{\Sigma_0}\left(n - i + \frac{\omega}{1-\omega}\right) \qquad \text{for } i = 1, 2, \ldots, n; \alpha = \beta,$$

$$(5.8)\quad q(n \mid i) = \frac{1}{\Sigma_0}\left(i + \frac{\rho}{1-\rho}\right) \qquad \text{for } i = 0, 1, \ldots, n-1; \alpha = \beta,$$

where Σ, and Σ_0 are given in (5.4). Obviously

$$q(0 \mid i) + q(n \mid i) = 1 \qquad \text{for } i = 1, 2, \ldots, n-1,$$
$$q(0 \mid 0) = \rho q(n \mid 1) \text{ and } \quad q(0 \mid n) = \omega q(0 \mid n-1).$$

We see that with the appropriate change of notation, (5.6)–(5.7) agree with the results of [26] in the special case $\rho = \omega$, $\gamma = 0$, and $i = 1, 2, \ldots, n-1$ (cf. [11, pp. 344–349], [13, p. 108] for $i = 1, 2, \ldots, n$, $\rho = \omega = \gamma = 0$, and [9]).

Let t_i be the mean time to absorption, given the starting state i, $i = 0, 1, 2, \ldots, n$. Obviously, the ith component of Me, e being the column vector of length $n+1$ with all entries equal to 1, is the mean time to absorption given by

$$t_0 = \frac{\alpha}{(\alpha-\rho\beta)\Sigma}\begin{Bmatrix} \left[1 + \frac{\rho}{\alpha}\left[n - \frac{\alpha}{\alpha-\beta}\left(\frac{\beta-\omega\alpha}{\beta(1-\omega)}\right)\right]\right]\left(\frac{\alpha}{\beta}\right)^n \\ -\frac{\beta-\omega\alpha}{\beta(1-\omega)}\left(1 - \frac{\rho}{\alpha-\beta}\right) \end{Bmatrix} \qquad \text{for } \alpha \neq \beta,$$

$$(5.9)\qquad = \frac{1}{(1-\rho)\Sigma_0}\left\{\left[1 + \frac{\rho}{2\alpha}\left[n - 1 + \frac{2\omega}{1-\omega}\right]\right]n + \frac{\omega}{1-\omega}\right\} \qquad \text{for } \alpha = \beta,$$

$$t_i = \frac{1}{\alpha-\beta}\begin{Bmatrix} \frac{1}{\Sigma}\left(\frac{\alpha}{\beta}\right)^{n-i}\left[n + \frac{\beta\rho}{\alpha(1-\rho)} + \frac{\omega\alpha}{\beta(1-\omega)}\right] \\ \left[\left(\frac{\alpha}{\beta}\right)^i - \frac{1-\rho}{1-\rho/\delta}\right] - i - \frac{\rho\beta}{\alpha(1-\rho)} \end{Bmatrix} \qquad \text{for } \alpha \neq \beta,$$

$$(5.10)\quad = \frac{1}{2\beta}\left\{\frac{1}{\Sigma_0}\begin{Bmatrix} \left[n\left(i + \frac{\rho}{1-\rho}\right) + \frac{\rho}{1-\rho}\right] \\ \left(n + \frac{2\omega}{1-\omega}\right) + i\left(\frac{\omega}{1-\omega} - \frac{\rho}{1-\rho}\right) \end{Bmatrix} - i^2\right\} \qquad \text{for } \alpha = \beta,$$

for $i = 1, 2, \ldots, n-1$, and

$$t_n = \frac{1}{(1-\omega)\Sigma}\begin{Bmatrix} \left(1 + \frac{\omega}{\alpha-\beta}\right)\left(\frac{\alpha}{\beta}\right)^n - \frac{\alpha(1-\rho)}{\alpha-\beta\rho} \\ \left(\frac{\omega}{\beta}n + 1\right) - \frac{\omega}{\alpha-\beta} \end{Bmatrix} \qquad \text{for } \alpha \neq \beta,$$

$$(5.11)\quad = \frac{1}{(1-\omega)\Sigma_0}\left\{\left[1 + \frac{\omega}{2\beta}\left[n - 1 + \frac{2\rho}{1-\rho}\right]\right]n + \frac{\rho}{1-\rho}\right\} \qquad \text{for } \alpha = \beta,$$

where Σ and Σ_0 are given in (5.4).

We see that the expressions (5.9)–(5.11) with the appropriate change of notation agree with the well-known results for a random walk between two perfectly absorbing barriers, $\rho = \omega = 0$, in the case $\gamma = 0$ (cf. [11, p. 348], [13, p. 108], [26], and [9]).

6. Conclusions. In the present paper a detailed connection between tridiagonal matrices, three-term recurrence relations, and the Markov chain random walk model is given. Closed form expressions for the entries of the inverse of the tridiagonal matrix $P = P(q, r, p)$ with $q = (q_0, q_1, \ldots, q_n)$, $r = (r_0, r_1, \ldots, r_n)$, and $p = (p_0, p_1, \ldots, p_n)$, via the Cholesky decompositional approach, in terms of the determinants of tridiagonal submatrices of P, are introduced. These determinants are solutions of a linear second-order homogeneous difference equation with variable coefficients. The presented results provide a simple and convenient approach to finding explicit expressions for the fundamental matrix of the model under investigation, with some interesting particular cases**.** In particular, the results are applied to MC-RW problems, in the presence of two semiabsorbing boundaries at 0 and n. Very simple expressions for the entries of the fundamental matrix are also given. This allows us to obtain general expressions for the mean of the occupation total explicitly in terms of the reflection probabilities ρ, ω and the basic transition probabilities $p_{ij}, i, j = 0, 1, \ldots, n$. These expressions generalize the results of [13, p. 100] to the case ρ, ω and γ nonzero. Exact analytical expressions for the absorption probabilities at the boundaries are also obtained. These expressions generalize the results of [9] to the case γ nonzero and the results of [26] to the case $\rho \neq \omega, \gamma \neq 0$ and $i = 0, 1, \ldots, n$.

Appendix. Proof of Corollary 3.1. Equation (3.4) can be reduced from second order to first order:

$$Y_m = p_m Y_{m-1} + (1 - r_0 - p_0) \prod_{j=1}^{m} q_j, m \geq 1. \tag{A.1}$$

One may obtain the solution of the associated homogeneous equation of (A.1) by a simple iteration:

$$Y_m = (1 - r_0) \prod_{j=1}^{m} p_j. \tag{A.2}$$

The unique solution of the nonhomogeneous equation (A.1) may be found by dividing both the left- and right-hand sides of (A.1) by $\prod_{j=1}^{m} p_j$, which can be written as

$$\Delta Y_m \left(\prod_{j=1}^{m} p_j \right)^{-1} = (1 - r_0 - p_0) \prod_{j=1}^{m+1} \frac{q_j}{p_j},$$

where Δ is the first difference operator. Therefore, a particular solution of (A.1) is

$$Y_m \left(\prod_{j=1}^{m} p_j \right)^{-1} = \Delta^{-1} \left((1 - r_0 - p_0) \prod_{j=1}^{m+1} \frac{q_j}{p_j} \right),$$

which can be expressed as

$$Y_m = (1 - r_0 - p_0) \prod_{j=1}^{m} p_j \sum_{u=1}^{m} \prod_{j=1}^{u} \frac{q_j}{p_j}. \tag{A.3}$$

From (A.2) and (A.3), the general solution of (A.1) is given as (3.6). Formula (3.7) can be proven in a completely analogous manner.

Acknowledgment. I am grateful to a referee for his valuable comments and helpful remarks which helped to improve the final version of this paper.

REFERENCES

[1] F. Ahmed, *A system of equations with a tridiagonal coefficient matrix*, Appl. Math. Comp., 159 (2004), pp. 435–438.

[2] M. Adivar and E. Bairamov, *Difference equations of second order with spectral singularities*, J. Math. Anal. Appl., 277 (2003), pp. 714–721.

[3] F. G. Boese, *On ordinary difference equations with variable coefficients*, J. Math. Anal. Appl., 273 (2002), pp. 378–408.

[4] P. Coolen-Schrijner, and E. A. Van Doorn, *Analysis of random walks using orthogonal polynomials*, J. Comput. Appl. Math., 99 (1998), pp. 387–399.

[5] D. R. Cox and H. D. Miller, *The Theory of Stochastic Processes*, Methuen, London, 1965.

[6] C. M. Da Fonseca and J. Petronilho, *Explicit inverses of some tridiagonal matrices*, Linear Algebra Appl., 325 (2001), pp. 7–21.

[7] S. N. Elaydi, *An Introduction to Difference Equations*, 2nd ed., Springer-Verlag, New York, 1999.

[8] M. A. El-Shehawey, *On the frequency count for a random walk with absorbing boundaries: A carcinogenesis example.* I., J. Phys. A: Math. Gen., 27 (1994), pp. 7035–7046.

[9] M. A. El-Shehawey, *Absorption probabilities for a random walk between two partially absorbing boundaries.* I., J. Phys. A: Math. Gen., 33 (2000), pp. 9005–9013.

[10] D. Fasino and L. Gemignani, *Structure and computational properties of possibly singular semiseparable matrices*, Linear Algebra Appl., 340 (2002), pp. 183–198.

[11] W. Feller, *An Introduction to Probability Theory and its Applications*, Vol. 1, 3rd ed., John Wiley & Sons, New York, 1968.

[12] B. D. Hughes, *Random Walks and Random Environments*, Vol. 1. Random Walks, Oxford University, The Clarendon Press, New York, 1996.

[13] M. Iosifescu, *Finite Markov Processes and their Applications*, John Wiley and Sons, Chichester, 1980.

[14] J. W. Lewis, *Inversion of tridiagonal matrices*, Numer. Math., 38 (1982), pp. 333–345.

[15] L.-Z. Lu, W.-K. Ching, and M. K. Ng, *Exact algorithms for singular tridiagonal systems with applications to Markov chains*, Appl. Math. Comput., 159 (2004), pp. 275–289.

[16] R. K. Mallik, *On the solution of a second order linear homogeneous difference equation with variable coefficients*, J. Math. Anal. Appl., 215 (1997), pp. 32–47.

[17] R. K. Mallik, *The inverse of a lower triangular matrix*, Int. J. Math., Game Theory, Algebra, 8 (1999), pp. 167–174.

[18] R. K. Mallik, *On the solution of a linear homogeneous difference equation with variable coefficients*, SIAM J. Math. Anal., 31 (2000), pp. 375–385.

[19] R. K. Mallik, *The inverse of a tridiagonal matrix*, Linear Algebra Appl., 325 (2001), pp. 109–139.

[20] G. Meurant, *A review on the inverse of symmetric tridiagonal and block tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 707–728.

[21] J. G. Mauldon, *A matrix with application to random walk, Brownian motion, and ring theory*, Amer. Math. Monthly, 94 (1987), pp. 423–436.

[22] M. F. Neuts, *General transition probabilities for finite Markov chains*, Proc. Cambridge Philos. Soc., 60 (1964), pp. 83–91.

[23] J. R. Norris, *Markov Chains*, Cambridge University Press, Cambridge, 1997.

[24] A. M. Odlyzko, *Linear recurrences with varying coefficients*, in Handbook of Combinatorics, Vol. 2, R. L. Graham, M. Grötschel, and L. Lovasz, eds., Elsevier, Amsterdam, 1995, pp. 1135–1138.

[25] R. Peluso and T. Politi, *Some improvements for two-sided bounds on the inverse of diagonally dominant tridiagonal matrices*, Linear Algebra Appl., 330 (2001), pp. 1–14.

[26] O. E. Percus, *Phase transition in one-dimensional random walk with partially reflecting boundaries*, Adv. in Appl. Prob., 17 (1985), pp. 594–606.

[27] G. Rudolph, *The fundamental matrix of the general random walk with absorbing boundaries*, Technical Report of the Collaborative Research Center "Computational Intelligence" CI-75, University of Dortmund, Dortmund, Germany, 1999, pp. 1–5.

[28] S. K. Srinivasan and K. M. Mehata, *Stochastic Processes*, McGraw-Hill, New Delhi, 1976.

[29] R. Usmani, *Inversion of a tridiagonal Jacobi matrix*, Linear Algebra Appl., 212 (1994), pp. 413–414.

[30] J. Wittenburg, *Inverses of tridiagonal Toeplitz and periodic matrices with applications to mechanics*, J. Appl. Math. Mech., 62 (1998), pp. 575–587.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 509–521

DESCRIPTION OF SYMMETRIC AND SKEW–SYMMETRIC SOLUTION SET*

MILAN HLADÍK†

Abstract. We consider a linear system $Ax = b$, where A is varying inside a given interval matrix $\mathbf{A}$, and b is varying inside a given interval vector $\mathbf{b}$. The solution set of such a system is described by the well-known Oettli–Prager Theorem. But if we are restricted only on symmetric/skew–symmetric matrices $A \in \mathbf{A}$, the problem is much more complicated. So far, the symmetric/skew–symmetric solution set description could be obtained only by a lengthy Fourier–Motzkin elimination applied on each orthant. We present an explicit necessary and sufficient characterization of the symmetric and skew–symmetric solution set by means of nonlinear inequalities. The number of the inequalities is, however, still exponential w.r.t. the problem dimension.

Key words. linear interval systems, solution set, interval matrix, symmetric matrix

AMS subject classifications. 65G40, 15A06, 15A57

DOI. 10.1137/070680783

NOTATION

$\mathbb{IR}^{m\times n}$	the set of all m-by-n interval matrices
$\mathbb{IR}^n$	the set of all n-dimensional interval vectors
$\square S$	interval hull of a set $S \subset \mathbb{R}^n$, i.e., the smallest box $[a_1, b_1] \times \cdots \times [a_n, b_n]$ that contains all the elements of S
$\prec_{\text{lex}}$	strict lexicographic ordering of vectors, i.e., $u \prec_{\text{lex}} v$ if for some k we have $u_i = v_i$, $i < k$, and $u_k < v_k$
$\preceq_{\text{lex}}$	lexicographic ordering of vectors, i.e., $u \preceq_{\text{lex}} v$ if $u \prec_{\text{lex}} v$ or $u = v$
$\lvert v\rvert$	absolute value of a vector v, i.e., the vector with components $\lvert v\rvert_i = \lvert v_i\rvert$
$A_{i,\bullet}$	the ith row of a matrix A
e_k	the kth basis vector (with convenient dimension), i.e., the kth column of the identity matrix
r^+	positive part of a real number r, i.e., $r^+ = \max(0, r)$

1. Introduction.

Real-life problems are often subject to uncertainties in data measurements. Such uncertainties can be dealt with by methods of interval analysis [1] instead of exact values we compute with compact real intervals. An interval matrix is defined as

$$\boldsymbol{A} = [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m\times n} \mid \underline{A} \le A \le \overline{A}\},$$

where $\underline{A} \le \overline{A}$ are fixed matrices (n-dimensional interval vectors can be regarded as interval matrices n-by-1). By

$$A^c \equiv \frac{1}{2}(\underline{A} + \overline{A}), \quad A^\Delta \equiv \frac{1}{2}(\overline{A} - \underline{A})$$

we denote the midpoint and radius of $\boldsymbol{A}$, respectively.

*Received by the editors January 23, 2007; accepted for publication (in revised form) by A. Frommer February 12, 2008; published electronically May 9, 2008.
http://www.siam.org/journals/simax/30-2/68078.html
†Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Malostranské nám. 25, 118 00, Prague, Czech Republic (milan.hladik@matfyz.cz).

Let us consider a system of linear interval equations

$$\boldsymbol{A}x = \boldsymbol{b}.$$

The solution set

$$\Sigma \equiv \{x \in \mathbb{R}^n \mid Ax = b,\ A \in \boldsymbol{A},\ b \in \boldsymbol{b}\}$$

is described by the well-known Oettli–Prager condition [11]

$$x \in \Sigma \quad \Leftrightarrow \quad A^{\Delta}|x| + b^{\Delta} \geq |A^c x - b^c|.$$

In interval analysis, we usually suppose, that values vary in given intervals independently. But in some applications, dependencies can occur (cf. [5], [9]). Especially, we focus on some types of the matrix A. The symmetric solution set is defined as

$$\Sigma_{sym} \equiv \{x \in \mathbb{R}^n \mid Ax = b,\ A = A^T,\ A \in \boldsymbol{A},\ b \in \boldsymbol{b}\},$$

and the skew–symmetric solution set as

$$\Sigma_{skew} \equiv \{x \in \mathbb{R}^n \mid Ax = b,\ A = -A^T,\ A \in \boldsymbol{A},\ b \in \boldsymbol{b}\}.$$

These sets have been exhaustively studied in recent years (see [2], [3], [4], [5], [6], and [7]). Applications involve Markov chains [8] and truss mechanics [10], for instance. Descriptions of Σ_{sym} and Σ_{skew} can be obtained by a Fourier–Motzkin elimination applied on each of 2^n orthants. Contrary to Σ, the symmetric solution set Σ_{sym} is not polyhedral, its shape is described by quadrics (see [3], [4], [5], and [6]), and it is not convex in general, even if intersected with an orthant.

The paper is organized as follows. In section 2 we derive a solution set characterization for a system of linear interval equations, where specific dependences occur. As consequences, we obtain a description of the symmetric solution set Σ_{sym} (section 3), and a description of the skew–symmetric solution set Σ_{skew} (section 4). The basic properties of Σ_{sym}, which were mentioned above, simply follow from the proposed Theorem 3.1 in section 3 (illustrated by Figures 3.1 and 3.2).

2. Linear interval equations with particular dependences. This section provides a characterization of the linear interval system equipped with a certain dependency (Theorem 2.2); the matrix A occurs twice in the system—in (2.3) and transposed in (2.4). We will see later in sections 3 and 4 that the description of the symmetric/skew–symmetric solution set is a simple consequence of Theorem 2.2. Another reason for dealing with such a dependency is that similar relations (occurrence of a matrix and its transposition in a system) can appear in some applications, e.g., optimality conditions in linear programming.

First we state an auxiliary result.

LEMMA 2.1. [illegible] $a^1, b^1, d^1 \in \mathbb{R}^m$, $a^2, b^2, d^2 \in \mathbb{R}^n$, [illegible] $C \in \mathbb{R}^{m \times n}$ [illegible]

$$f(u,v) \equiv (a^1)^T u + (b^1)^T |u| + (a^2)^T v + (b^2)^T |v| + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} |d_j^2 u_i + d_i^1 v_j| \tag{2.1}$$

[illegible] $u \in \mathbb{R}^m$ [illegible] $v \in \mathbb{R}^n$ [illegible] u, v [illegible]

(i) $u_i \in \{0, d_i^1\}\ \forall\ i = 1, \ldots, m$, [illegible] $v_j \in \{0, -d_j^2\}\ \forall\ j = 1, \ldots, n$
(ii) $u_i \in \{0, -d_i^1\}\ \forall\ i = 1, \ldots, m$, [illegible] $v_j \in \{0, d_j^2\}\ \forall\ j = 1, \ldots, n$
(iii) $(u^T, v^T)^T = \pm e_k$ [illegible] $k \in \{1, \ldots, m+n\}$

. One implication is obvious: If $f(u,v)$ is nonnegative for all $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$, then it is nonnegative in particular points.

The converse will be proven by induction w.r.t. the dimension $m+n$.

If $m+n=1$, then without loss of generality (w.l.o.g.) assume $m=1$, $n=0$. The function $f(u)=a^1u+b^1|u|$ is nonnegative for all real u iff it is nonnegative for $u=\pm 1$, which is managed by the third condition of Lemma 2.1.

The induction step will be proven by contradiction. Let us assume that $f(u,v)<0$ for some vectors u,v.

(a) Suppose there are some vectors u,v such that $f(u,v)<0$ and $u_i=0$ for some index i. Delete the ith component in u and denote the resulting vector by $\tilde{u}$. Replace b^2 by $\tilde{b}^2$, where $\tilde{b}^2_j = b^2_j + c_{ij}|d^1_i|$, $j=1,\dots,n$, and apply the induction hypothesis to $\tilde{u},v$. Hence $f(\tilde{u}',v')\geq 0$ for some vectors $\tilde{u}'\in\mathbb{R}^{m-1}$ and $v'\in\mathbb{R}^n$ satisfying one of the conditions (i)–(iii). Canonical embedding of $\tilde{u}'$ to the space $\mathbb{R}^m$ yields the pair of vectors $u'\in\mathbb{R}^m$ and $v'\in\mathbb{R}^n$ such that $f(u',v')\geq 0$, and one of the conditions (i)–(iii) is true. Thus, a contradiction.

(b) Suppose there are some vectors u,v such that $f(u,v)<0$ and $v_j=0$ for some index j. Here the assertion follows analogously to case (a).

(c) Assume—as the remaining case—that no component of u,v is zero for all vectors u,v with $f(u,v)<0$.

First we show that $d^1_i\neq 0$ for every $i=1,\dots,m$, and $d^2_j\neq 0$ for every $j=1,\dots,n$. If w.l.o.g. $d^1_i=0$ for some i, then we have

$$f(u,v)=f(u_1,\dots,u_{i-1},0,u_{i+1},\dots,u_m,v)+f(0,\dots,0,u_i,0,\dots,0,0)<0.$$

That is, one of the two summands is negative, which contradicts our assumption.

Now, choose vectors $\tilde{u},\tilde{v}$ with $f(\tilde{u},\tilde{v})<0$ such that the number of absolute values in (2.1) that are zero is maximal. Define the graph $G=(V,E)$, where the vertex set V consist of $\tilde{u}_i$, $i=1,\dots,m$, and $\tilde{v}_j$, $j=1,\dots,n$. The edge set E contains such pairs $\{\tilde{u}_i,\tilde{v}_j\}$ for which $d^2_j\tilde{u}_i + d^1_i\tilde{v}_j = 0$. We distinguish three cases and show that each of them contradicts some assumption.

1. The graph G is connected. Choose $(\tilde{u}_{i^*},\tilde{v}_{j^*})\in E$ and define $z^*\equiv -\frac{\tilde{v}_{j^*}}{d^2_{j^*}}\neq 0$. Then $\tilde{u}_{i^*}=d^1_{i^*}z^*$, and $\tilde{v}_{j^*}=-d^2_{j^*}z^*$. Due to the connectivity of G, we can extend this property by induction to all i,j: If $(\tilde{u}_i,\tilde{v}_j)\in E$ and $\tilde{u}_i=d^1_iz^*$, then $\tilde{v}_j=-d^2_jz^*$. If $(\tilde{u}_i,\tilde{v}_j)\in E$ and $\tilde{v}_j=-d^2_jz^*$, then $\tilde{u}_i=d^1_iz^*$. Hence

$$\tilde{u}_i = d^1_iz^* \ \ \forall\ i=1,\dots,m, \quad \tilde{v}_j=-d^2_jz^*\ \forall\ j=1,\dots,n. \tag{2.2}$$

Define $u'\equiv\frac{1}{|z^*|}\tilde{u}$, $v'\equiv\frac{1}{|z^*|}\tilde{v}$. Vectors u',v' satisfy the first or the second condition of Lemma 2.1 (depending on the sign of z^*), but $f(u',v')=\frac{1}{|z^*|}f(u,v)<0$. Thus, a contradiction.

2. The graph G is not connected and $E\neq\emptyset$. We will construct vectors u',v' with $f(u',v')<0$ and at least one component of u' or v' to be zero, which contradicts the assumption of case (c).

Take a connected component $G'=(V',E')$ of G such that $E'\neq\emptyset$. Then the property (2.2) holds when restricted on G':

$$\tilde{u}_i = d^1_iz^* \ \ \forall\ i:\tilde{u}_i\in V', \quad \tilde{v}_j=-d^2_jz^*\ \ \forall\ j:\tilde{v}_j\in V'.$$

Consider the function $g(z) \equiv f(u(z), v(z))$ as a function of variable z, where

$$u_i(z) = \begin{cases} d_i^1 z & \tilde{u}_i \in V', \\ \tilde{u}_i & \text{otherwise}, \end{cases} \qquad v_j(z) = \begin{cases} -d_j^2 z & \tilde{v}_j \in V', \\ \tilde{v}_j & \text{otherwise}. \end{cases}$$

Then $g(z)$ is a piecewise linear function (broken line) on $\mathbb{R}$. Moreover, it is linear on a neighborhood $N(z^*)$ of z^*, that is, $g(z) = pz + q$, $z \in N(z^*)$ for some $p, q \in \mathbb{R}$. W.l.o.g. assume that $z^* > 0$ and consider two possibilities.

Let $g(z)$ be nondecreasing on $N(z^*)$. In this case $g(z)$ is nondecreasing on the interval $[0, z^*]$, since otherwise there is a break point in $(0, z^*)$ contradicting our assumption on the maximal number of zero absolute values. From $g(0) \le g(z^*) = f(\tilde{u}, \tilde{v}) < 0$ we get $g(0) = f(u(0), v(0)) < 0$ with $u(0)_i = 0$ for all indices i such that $\tilde{u}_i \in V'$ (at least one exists due to $E' \neq \emptyset$). This contradicts the assumption of case (c).

Let $g(z)$ be decreasing on $N(z^*)$. Then $g(z)$ is decreasing on $[z^*, \infty)$ (otherwise we are in contradiction with our assumption on the maximal number of zero absolute values). Moreover, for sufficiently large z we have $f(u', v') < 0$, where

$$u_i' = \begin{cases} d_i^1 z & \tilde{u}_i \in V', \\ 0 & \text{otherwise}, \end{cases} \qquad v_j' = \begin{cases} -d_j^2 z & \tilde{v}_j \in V', \\ 0 & \text{otherwise}. \end{cases}$$

As $V' \subsetneqq V$, the vectors u', v' contradict the assumption of case (c).

3. The graph G is not connected and $E = \emptyset$. Define the function $g(z) \equiv f(\tilde{u}_1 z, \tilde{u}_2, \dots, \tilde{u}_m, \tilde{v}_1, \dots, \tilde{v}_n)$. This function is linear on a neighborhood $N(z^*)$ of $z^* \equiv 1$.

If $g(z)$ is nondecreasing on $N(z^*)$, then it is nondecreasing on $[0, z^*]$ (otherwise we are in contradiction with our assumption on the maximal number of zero absolute values). From $g(0) \le g(z^*) = f(\tilde{u}, \tilde{v}) < 0$ we get $g(0) = f(0, \tilde{u}_2, \dots, \tilde{u}_m, \tilde{v}_1, \dots, \tilde{v}_n) < 0$. This contradicts the assumption of case (c).

If $g(z)$ is decreasing on $N(z^*)$, then it is decreasing on $[z^*, \infty)$ (otherwise we are in contradiction with our assumption on the maximal number of zero absolute values). Moreover, for sufficiently large z we have $f(\tilde{u}_1 z, 0, \dots, 0, 0, \dots, 0) < 0$. This also contradicts the assumption of case (c). □

THEOREM 2.2. [illegible] $\boldsymbol{A} \in \mathbb{IR}^{n \times n}$, $\boldsymbol{b} \in \mathbb{IR}^n$, [illegible] $\boldsymbol{d} \in \mathbb{IR}^n$ [illegible] $x, y \in \mathbb{R}^n$ [illegible]

$$Ax = b, \tag{2.3}$$

$$A^T y = d \tag{2.4}$$

[illegible] $A \in \boldsymbol{A}$, $b \in \boldsymbol{b}$, [illegible] $d \in \boldsymbol{d}$. [illegible]

$$A^\Delta |x| + b^\Delta \ge |r^1|, \tag{2.5}$$

$$A^\Delta |y| + d^\Delta \ge |r^2|, \tag{2.6}$$

$$\sum_{i,j=1}^n a_{ij}^\Delta |y_i x_j (p_i - q_j)| + \sum_{i=1}^n (b_i^\Delta |y_i p_i| + d_i^\Delta |x_i q_i|) \ge \left| \sum_{i=1}^n (r_i^1 y_i p_i - r_i^2 x_i q_i) \right| \tag{2.7}$$

$\forall\, p, q \in \{0,1\}^n$, [illegible] $r^1 \equiv -A^c x + b^c$, $r^2 \equiv -(A^c)^T y + d^c$

. Let $x, y \in \mathbb{R}^n$. Then x, y satisfy (2.3)–(2.4) iff for a certain $\alpha \in [-1,1]^{n\times n}$ the following relations hold:

$$A^c_{i,\bullet}x + \sum_{k=1}^{n} \alpha_{ik} a^\Delta_{ik} x_k \in [b^c_i - b^\Delta_i, b^c_i + b^\Delta_i] \ \forall\ i = 1,\dots,n,$$
$$\left(A^c_{\bullet,j}\right)^T y + \sum_{k=1}^{n} \alpha_{kj} a^\Delta_{kj} y_k \in [d^c_j - d^\Delta_j, d^c_j + d^\Delta_j] \ \forall\ j = 1,\dots,n.$$

Equivalently, iff the following linear programming problem

$$\max \sum_{i,j=1}^{n} 0 \cdot \alpha_{ij}$$

subject to

$$-\sum_{k=1}^{n} \alpha_{ik} a^\Delta_{ik} x_k \le -r^1_i + b^\Delta_i \ \forall\ i = 1,\dots,n,$$
$$\sum_{k=1}^{n} \alpha_{ik} a^\Delta_{ik} x_k \le r^1_i + b^\Delta_i \ \forall\ i = 1,\dots,n,$$
$$-\sum_{k=1}^{n} \alpha_{kj} a^\Delta_{kj} y_k \le -r^2_j + d^\Delta_j \ \forall\ j = 1,\dots,n,$$
$$\sum_{k=1}^{n} \alpha_{kj} a^\Delta_{kj} y_k \le r^2_j + d^\Delta_j \ \forall\ j = 1,\dots,n,$$
$$\alpha_{ij} \le 1 \ \forall\ i,j = 1,\dots,n,$$
$$-\alpha_{ij} \le 1 \ \forall\ i,j = 1,\dots,n$$

has an optimal solution.

Recall duality in linear programming [12], [13]. The linear programs

$$\max \tilde{b}^T \tilde{y} \ \text{ subject to } \ \tilde{A}^T \tilde{y} \le \tilde{c}$$

and

$$\min \tilde{c}^T \tilde{x} \ \text{ subject to } \ \tilde{A}\tilde{x} = \tilde{b}, \tilde{x} \ge 0$$

are dual to each other. Moreover, their optimal values are equal as long as at least one of the problems is feasible (i.e., the constraints are satisfiable).

Thus our linear programming problem has an optimal solution iff the dual problem

$$\min \ \Big\{(-r^1 + b^\Delta)^T w^1 + (r^1 + b^\Delta)^T w^2 + (-r^2 + d^\Delta)^T w^3 + \ (r^2 + d^\Delta)^T w^4 + \sum_{i,j=1}^{n} (w^5_{ij} + w^6_{ij})\Big\}$$

subject to

$$-a^\Delta_{ij} x_j w^1_i + a^\Delta_{ij} x_j w^2_i - a^\Delta_{ij} y_i w^3_j + a^\Delta_{ij} y_i w^4_j + w^5_{ij} - w^6_{ij} = 0 \ \forall\ i,j = 1,\dots,n,$$
$$w^1, w^2, w^3, w^4, w^5, w^6 \ge 0$$

has an optimal solution. The dual problem is feasible as its constraints are fulfilled when all the variables are equal to zero, for instance. After substitution $u \equiv w^2 - w^1$,

$v \equiv w^4 - w^3$ we can rewrite this problem:

$$\min\ (r^1 + b^\Delta)^T u + 2(b^\Delta)^T w^1 + (r^2 + d^\Delta)^T v + 2(d^\Delta)^T w^3 + \sum_{i,j=1}^{n} (w^5_{ij} + w^6_{ij})$$

subject to

$$\begin{aligned} a^\Delta_{ij} x_j u_i + a^\Delta_{ij} y_i v_j + w^5_{ij} - w^6_{ij} = 0 \ \forall\ i,j = 1,\dots,n,\\ w^1 \geq -u,\\ w^3 \geq -v,\\ w^1, w^3, w^5, w^6 \geq 0. \end{aligned}$$

For w^1, w^3, w^5, and w^6 some necessary optimality conditions can be given.

For each i,j at least one of w^5_{ij},w^6_{ij} is zero (otherwise subtract from them a sufficiently small $\varepsilon > 0$ and obtain a better solution). If $w^5_{ij} = 0$, then $w^6_{ij} = a^\Delta_{ij} x_j u_i + a^\Delta_{ij} y_i v_j \geq 0$, and hence $w^5_{ij} + w^6_{ij} = |a^\Delta_{ij} x_j u_i + a^\Delta_{ij} y_i v_j|$. Similarly, if $w^6_{ij} = 0$, then $w^5_{ij} = -(a^\Delta_{ij} x_j u_i + a^\Delta_{ij} y_i v_j) \geq 0$, and hence $w^5_{ij} + w^6_{ij} = |a^\Delta_{ij} x_j u_i + a^\Delta_{ij} y_i v_j|$. Therefore

$$w^5_{ij} + w^6_{ij} = |a^\Delta_{ij} x_j u_i + a^\Delta_{ij} y_i v_j|$$

holds in any case.

Next, the only constraints involving the variable w^1_i, $i \in \{1,\dots,n\}$ are $w^1_i \geq -u_i$ and $w^1_i \geq 0$. Since the objective function coefficient by w^1_i is nonnegative, the optimal w^1_i should be as small as possible. That is, $w^1_i = \max(-u_i, 0) = (-u_i)^+$. Hence we have $w^1 = (-u)^+$, and the equation $w^3 = (-v)^+$ follows analogously.

Using these necessary optimality conditions, the optimization problem can be reformulated as an unconstrained optimization problem:

$$\begin{aligned} \min_{u,v\in\mathbb{R}^n}\ \Big\{ &(r^1 + b^\Delta)^T u + 2(b^\Delta)^T (-u)^+ + (r^2 + d^\Delta)^T v \\ &+ 2(d^\Delta)^T (-v)^+ + \sum_{i,j=1}^{n} a^\Delta_{ij} |x_j u_i + y_i v_j| \Big\}. \end{aligned}$$

The positive part of a real number p is equal to $p^+ = \frac{1}{2}(p + |p|)$, and the problem comes in the form

$$\min_{u,v\in\mathbb{R}^n}\ (r^1)^T u + (b^\Delta)^T |u| + (r^2)^T v + (d^\Delta)^T |v| + \sum_{i,j=1}^{n} |a^\Delta_{ij} x_j u_i + a^\Delta_{ij} y_i v_j|.$$

As a^Δ_{ij} is nonnegative (because it is the radius of an interval), the objective function can be written

$$f(u,v) \equiv (r^1)^T u + (b^\Delta)^T |u| + (r^2)^T v + (d^\Delta)^T |v| + \sum_{i,j=1}^{n} a^\Delta_{ij} |x_j u_i + y_i v_j|. \tag{2.8}$$

Note that, it is positive homogeneous, that is,

$$f(\lambda u, \lambda v) = \lambda f(u,v) \ \forall\ \lambda \geq 0.$$

If $f(\tilde{u}, \tilde{v}) < 0$ for some vectors $\tilde{u}, \tilde{v} \in \mathbb{R}^n$, then $f(\lambda\tilde{u}, \lambda\tilde{v})$ tends to $-\infty$ for $\lambda \to \infty$, and the problem does not attain an optimum. On the other hand, if $f(u,v) \geq 0$ for

all $u, v \in \mathbb{R}^n$, then the optimal solution is $u = v = 0$. Thus the optimization problem has an optimal solution iff the objective function is nonnegative for all $u, v \in \mathbb{R}^n$.

We now use Lemma 2.1 with $a^1 \equiv r^1$, $a^2 \equiv r^2$, $b^1 \equiv b^\Delta$, $b^2 \equiv d^\Delta$, $C \equiv A^\Delta$, $d^1 \equiv y$, $d^2 \equiv x$, and $m = n$. It follows that it is sufficient to test nonnegativity of $f(u, v)$ for three cases:

1. $u_i \in \{0, y_i\}\ \forall\ i = 1, \dots, n$, and $v_j \in \{0, -x_j\}\ \forall\ j = 1, \dots, n$;
2. $u_i \in \{0, -y_i\}\ \forall\ i = 1, \dots, n$, and $v_j \in \{0, x_j\}\ \forall\ j = 1, \dots, n$;
3. $(u^T, v^T)^T = \pm e_k$ for some $k \in \{1, \dots, 2n\}$.

The first and second cases yield

$$\pm \sum_{i=1}^{n} r_i^1 y_i p_i + \sum_{i=1}^{n} b_i^\Delta |y_i p_i| \mp \sum_{i=1}^{n} r_i^2 x_i q_i + \sum_{i=1}^{n} d_i^\Delta |x_i q_i| + \sum_{i,j=1}^{n} a_{ij}^\Delta |y_i x_j (p_i - q_j)| \geq 0$$

or

$$\sum_{i,j=1}^{n} a_{ij}^\Delta |y_i x_j (p_i - q_j)| + \sum_{i=1}^{n} (b_i^\Delta |y_i p_i| + d_i^\Delta |x_i q_i|) \geq \left| \sum_{i=1}^{n} (r_i^1 y_i p_i - r_i^2 x_i q_i) \right|,$$

where $p, q \in \{0, 1\}^n$. In the third case when $u = \pm e_k$ and $v = 0$, we get

$$\pm r_k^1 + b_k^\Delta + \sum_{j=1}^{n} a_{kj}^\Delta |x_j| \geq 0,$$

which is the kth Oettli–Prager inequality in (2.5). Likewise $u = 0$, $v = \pm e_k$ yields the kth Oettli–Prager inequality in (2.6). □

3. Symmetric solution set. In this section, we suppose w.l.o.g. that $\boldsymbol{A} = \boldsymbol{A}^T$, i.e., matrices A^c, A^Δ are symmetric. Otherwise we restrict our considerations on the interval matrix $(\boldsymbol{a}_{ij} \cap \boldsymbol{a}_{ji})_{i,j=1}^n$.

Theorem 3.1, which is a simple corollary of Theorem 2.2, enables us to obtain an explicit description of the symmetric solution set Σ_{sym}. Nevertheless, the number of inequalities in the description is still exponential. Therefore when checking $x \in \Sigma_{sym}$ for only one vector x, it is better from the theoretical viewpoint to use the linear programming problem (from the proof of Theorem 2.2), which is polynomially solvable [13]. The question whether Σ_{sym} can be described by a polynomial number of inequalities is still open.

THEOREM 3.1. [illegible] $r \equiv -A^c x + b^c$ [illegible] Σ_{sym} [illegible]

[illegible]

$$A^\Delta |x| + b^\Delta \geq |r|, \tag{3.1}$$

$$\sum_{i,j=1}^{n} a_{ij}^\Delta |x_i x_j (p_i - q_j)| + \sum_{i=1}^{n} b_i^\Delta |x_i (p_i + q_i)| \geq \left| \sum_{i=1}^{n} r_i x_i (p_i - q_i) \right| \tag{3.2}$$

[illegible] $p, q \in \{0, 1\}^n \setminus \{0, 1\}$ [illegible]

$$p \prec_{\text{lex}} q \quad (p = 1 - q \vee \exists\ i : p_i = q_i = 0). \tag{3.3}$$

[illegible]. For every $A \in \boldsymbol{A}$, the matrix $\frac{1}{2}(A + A^T) \in \boldsymbol{A}$ is symmetric, and for every $b^1, b^2 \in \boldsymbol{b}$ we have $\frac{1}{2}(b^1 + b^2) \in \boldsymbol{b}$. Thus, Σ_{sym} can be equivalently described as the set of all $x \in \mathbb{R}^n$ satisfying

$$Ax = b^1, \tag{3.4}$$

$$A^T x = b^2 \tag{3.5}$$

for some $A \in \boldsymbol{A}$, $b^1, b^2 \in \boldsymbol{b}$. Put $y \equiv x$, $\boldsymbol{d} \equiv \boldsymbol{b}$ and apply Theorem 2.2 on system (3.4)–(3.5). We obtain that Σ_{sym} is described by (3.1)–(3.2) for all $p, q \in \{0,1\}^n$. To reduce the number of inequalities in (3.2), it is sufficient due to symmetry to consider only vectors $p, q \in \{0,1\}^n$ for which $p \preceq_{\text{lex}} q$. Obviously, the case $p = q$ is also redundant.

The inequality (3.2) corresponding to $p = 0$ and any $q \in \{0,1\}^n$ can be omitted for the following reason. Multiplying the Oettli–Prager system (3.1) by the vector $(|x_1 q_1|, \dots, |x_n q_n|)$ we obtain

$$\sum_{i,j=1}^{n} a_{ij}^{\Delta} |x_i x_j q_i| + \sum_{i=1}^{n} b_i^{\Delta} |x_i q_i| \geq \sum_{i=1}^{n} |r_i x_i q_i| \geq \left| \sum_{i=1}^{n} r_i x_i q_i \right|.$$

Due to the symmetry of A^{Δ} the first sum is equal to $\sum_{i,j=1}^{n} a_{ij}^{\Delta} |x_i x_j q_j|$, and hence the inequality

$$\sum_{i,j=1}^{n} a_{ij}^{\Delta} |x_i x_j q_j| + \sum_{i=1}^{n} b_i^{\Delta} |x_i q_i| \geq \left| \sum_{i=1}^{n} r_i x_i q_i \right|$$

is a consequence of the Oettli–Prager system.

The inequality (3.2) corresponding to any $p \in \{0,1\}^n$ and $q = 1$ is redundant as it is a consequence of the inequality (3.2) with $p' \equiv 1 - p$, $q' \equiv 0$ (which is redundant for the same reason as before); the right-hand sides of the inequalities are the same, and the left-hand side of the former inequality includes all of the left-hand side terms of the latter inequality and possibly some more positive terms.

Finally, we prove redundancy for all inequalities (3.2) with $p, q \in \{0,1\}^n \setminus \{0,1\}$, $p \prec_{\text{lex}} q$, and

$$p \neq 1 - q \text{ and } \forall\, i : (p_i = 1 \vee q_i = 1). \tag{3.6}$$

Clearly, (3.6) is equivalent to

$$\forall\, i : (p_i = 1 \vee q_i = 1) \text{ and } \exists\, i : p_i = q_i = 1. \tag{3.7}$$

Such an inequality is a consequence of the inequality (3.2) with $p' \equiv 1 - q$, $q' \equiv 1 - p$. The vectors p', q' satisfy the condition (3.3). □

We compute the number of inequalities for system (3.2).

PROPOSITION 3.2. [illegible] (3.2) [illegible] $\frac{1}{2}(4^n - 3^n - 2 \cdot 2^n + 3)$ [illegible]

[illegible] There are $(2^n - 2)^2$ pairs of vectors p, q satisfying $p, q \in \{0,1\}^n \setminus \{0,1\}$. Since for each pair p, q just one of the conditions $p \prec_{\text{lex}} q$, $p = q$, or $q \prec_{\text{lex}} p$ is true, the number of the vectors p, q satisfying $p, q \in \{0,1\}^n \setminus \{0,1\}$, $p \prec_{\text{lex}} q$, is equal to $\frac{1}{2}\big((2^n - 2)^2 - (2^n - 2)\big) = \frac{1}{2}(2^n - 2)(2^n - 3)$.

Now we focus on condition (3.7) which determines the "bad" cases. For every p, q define

$$I_{p,q} \equiv \{i = 1, \dots, n \mid p_i = q_i = 1\},\ J_{p,q} \equiv \{i = 1, \dots, n \mid p_i + q_i = 1\}.$$

Vectors $p, q \in \{0,1\}^n$ satisfy (3.7) iff $|I_{p,q}| \geq 1$ and $|I_{p,q}| + |J_{p,q}| = n$. The value $\binom{n}{k} 2^{n-k}$ identifies the number of $p, q \in \{0,1\}^n$ for which $|I_{p,q}| = k$ and $|J_{p,q}| = n - k$. Summing up for all $k = 1, \dots, n$ and using binomial expansion of $(1+2)^n$ we obtain the number of pairs $p, q \in \{0,1\}^n$ with property (3.7) is equal to

$$\binom{n}{1} 2^{n-1} + \binom{n}{2} 2^{n-2} + \cdots + \binom{n}{n} 2^0 = 3^n - 2^n.$$

From this amount we have to exclude the cases when $p = 1$ or $q = 1$:

$$3^n - 2^n - 2 \cdot (2^n - 1) + 1.$$

Exactly half of them satisfy $p \prec_{\text{lex}} q$. Eventually, we obtain the number in question:

$$\frac{1}{2}(2^n - 2)(2^n - 3) - \frac{1}{2}\big(3^n - 2^n - 2 \cdot (2^n - 1) + 1\big) = \frac{1}{2}(4^n - 3^n - 2 \cdot 2^n + 3). \qquad \square$$

The number of inequalities in (3.2) is exponential, but not as tremendous as by using Fourier–Motzkin elimination (no better upper bound is known than the double exponential one). Moreover, system (3.2) is characterized explicitly and is much more easy to handle.

Concretely, for $n = 2$ we have only one additional inequality (in comparison to two inequalities obtained by Fourier–Motzkin elimination [4]), for $n = 3$ this number rises up to 12 (cf. [3], [4], [6]; Fourier–Motzkin elimination leads to 44 inequalities).

3.3. For the two-dimensional case, the symmetric solution set is described by the system consisting of the Oettli–Prager inequalities (3.1)

$$a_{11}^{\Delta}|x_1| + a_{12}^{\Delta}|x_2| + b_1^{\Delta} \geq |-a_{11}^{c}x_1 - a_{12}^{c}x_2 + b_1^{c}|$$
$$a_{21}^{\Delta}|x_1| + a_{22}^{\Delta}|x_2| + b_2^{\Delta} \geq |-a_{21}^{c}x_1 - a_{22}^{c}x_2 + b_2^{c}|$$

supplemented by only one inequality (3.2)

$$a_{11}^{\Delta}x_1^2 + a_{22}^{\Delta}x_2^2 + b_1^{\Delta}|x_1| + b_2^{\Delta}|x_2| \geq |\,-a_{11}^{c}x_1^2 + a_{22}^{c}x_2^2 + b_1^{c}x_1 - b_2^{c}x_2|.$$

In the list below we mention some particular examples. Figures 3.1 and 3.2 illustrate a solution set (light gray color) and a symmetric solution set (gray color):

1. (Figure 3.1) $\boldsymbol{A} = \left(\begin{smallmatrix}[1,2] & [0,a]\\ [0,a] & -1\end{smallmatrix}\right)$, $\boldsymbol{b} = \left(\begin{smallmatrix}2\\2\end{smallmatrix}\right)$; here the interval hull $\square\Sigma$ can be arbitrarily larger than $\square\Sigma_{sym}$, depending on the real parameter $a > 0$.
2. (Figure 3.2) $\boldsymbol{A} = \left(\begin{smallmatrix}-1 & [-5,5]\\ [-5,5] & 1\end{smallmatrix}\right)$, $\boldsymbol{b} = \left(\begin{smallmatrix}1\\ [1,3]\end{smallmatrix}\right)$; here Σ is unbounded, but Σ_{sym} is bounded.

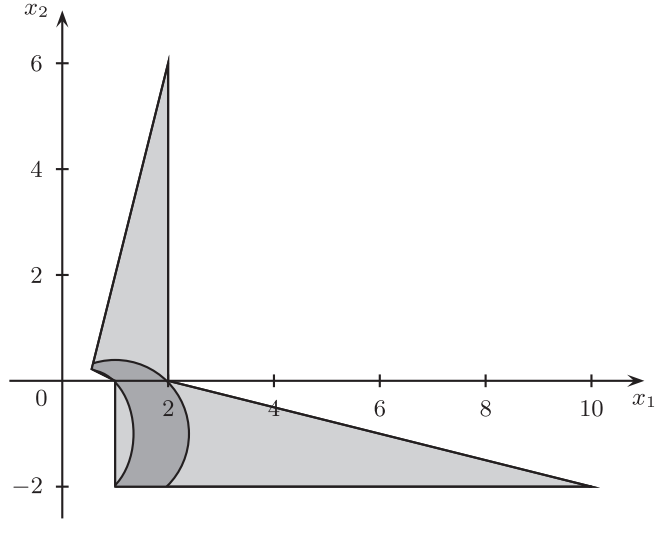


FIG. 3.1. *Solution set arbitrarily larger than symmetric solution set, $a = 4$.*

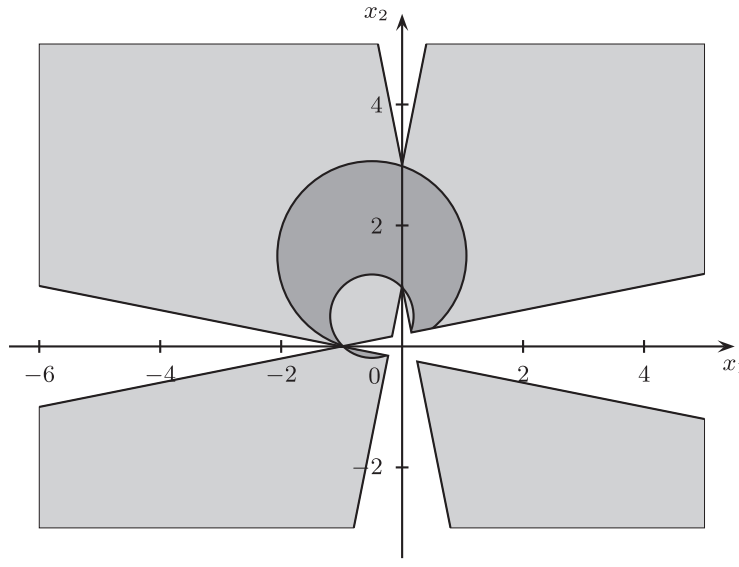


FIG. 3.2. *Unbounded solution set and bounded symmetric solution set.*

3. For $\boldsymbol{A} = \left(\begin{smallmatrix} [0,1] & [1,2] \\ [1,2] & [-1,0] \end{smallmatrix}\right)$, $\boldsymbol{b} = \left(\begin{smallmatrix} [-1,1] \\ [-1,1] \end{smallmatrix}\right)$ we have $\Sigma = \Sigma_{sym}$ and both are bounded.
4. For $\boldsymbol{A} = \left(\begin{smallmatrix} [-1,1] & [0,2] \\ [0,2] & [-1,1] \end{smallmatrix}\right)$, $\boldsymbol{b} = \left(\begin{smallmatrix} [0,1] \\ [0,1] \end{smallmatrix}\right)$ we have $\Sigma = \Sigma_{sym}$ and both are unbounded.

4. Skew–symmetric solution set. In this section, let us suppose w.l.o.g. that $\boldsymbol{A} = -\boldsymbol{A}^T$ and the diagonal of $\boldsymbol{A}$ is zero. Therefore A^c is skew–symmetric and A^Δ is a symmetric matrix. The description of the skew–symmetric solution set Σ_{skew} is a consequence of Theorem 2.2.

PROPOSITION 4.1. [illegible] $r \equiv -A^c x + b^c$ [illegible] Σ_{skew} [illegible]

$$A^\Delta |x| + b^\Delta \geq |r|, \tag{4.1}$$

$$\sum_{i,j=1}^{n} a_{ij}^\Delta |x_i x_j (p_i - q_j)| + \sum_{i=1}^{n} b_i^\Delta |x_i (p_i + q_i)| \geq \left| \sum_{i=1}^{n} r_i x_i (p_i + q_i) \right|$$

$$\forall\, p, q \in \{0,1\}^n \setminus \{0\},\ p \preceq_{\text{lex}} q. \tag{4.2}$$

[illegible]. For all $A \in \boldsymbol{A}$ and $b^1, b^2 \in \boldsymbol{b}$ we have that $\frac{1}{2}(A - A^T) \in \boldsymbol{A}$ is a skew–symmetric matrix and $\frac{1}{2}(b^1 + b^2) \in \boldsymbol{b}$. Thus, Σ_{skew} can be equivalently described as the set of all $x \in \mathbb{R}^n$ satisfying

$$Ax = b^1, \tag{4.3}$$

$$A^T(-x) = b^2 \tag{4.4}$$

for some $A \in \boldsymbol{A}$, $b^1, b^2 \in \boldsymbol{b}$. Put $y \equiv -x$, $\boldsymbol{d} \equiv \boldsymbol{b}$. Then

$$r^1 = -A^c x + b^c = -(-A^c)(-x) + b^c = -(A^c)^T y + d^c = r^2 \equiv r.$$

Apply Theorem 2.2 on system (4.3)–(4.4). We obtain that Σ_{skew} is described by (4.1)–(4.2). To reduce the number of inequalities in (4.2), it is sufficient due to symmetry to consider only vectors $p, q \in \{0,1\}^n \setminus \{0\}$ for which $p \preceq_{\text{lex}} q$. □

The number of inequalities in (4.2) is $2^{n-1}(2^n - 1)$ and can be furthermore decreased to the number $2^n - n - 1$; see Theorem 4.2 where we claim that it is sufficient to consider only such inequalities for which $p = q$, the others being redundant.

THEOREM 4.2. [illegible] $r \equiv -A^c x + b^c$ [illegible] Σ_{skew} [illegible]

$$A^{\Delta}|x| + b^{\Delta} \geq |r|, \tag{4.5}$$

$$\sum_{i<j} a_{ij}^{\Delta}|x_i x_j (p_i - p_j)| + \sum_{i=1}^{n} b_i^{\Delta}|x_i p_i| \geq \left|\sum_{i=1}^{n} r_i x_i p_i\right| \quad \forall\, p \in \{0,1\}^n \setminus \{0\},\ p \neq e_k. \tag{4.6}$$

[illegible]. For given vectors $p, q \in \{0,1\}^n$ denote the inequality (4.2) corresponding to p, q by $Ineq(p,q)$. Let p, q be fixed, and define vectors $s, t \in \{0,1\}^n$ componentwise by

$$s_i = \begin{cases} 1 & p_i = q_i = 1, \\ 0 & \text{otherwise}, \end{cases} \qquad t_i = \begin{cases} 1 & (p_i = 1) \vee (q_i = 1), \\ 0 & \text{otherwise}. \end{cases}$$

We prove that $Ineq(p,q)$ is a consequence of the inequality

$$\frac{1}{2}\big(Ineq(s,s) + Ineq(t,t)\big) \tag{4.7}$$

and hence can be omitted.

The right-hand side of the inequality $Ineq(p,q)$ is $|\sum_{p_i=1} r_i x_i + \sum_{q_i=1} r_i x_i| = |\sum_{s_i=1} r_i x_i + \sum_{t_i=1} r_i x_i|$, which is not greater than $|\sum_{s_i=1} r_i x_i| + |\sum_{t_i=1} r_i x_i|$, the right-hand side of (4.7). The second sum in $Ineq(p,q)$ is equal to $\sum_{p_i=1} b_i^{\Delta}|x_i| + \sum_{q_i=1} b_i^{\Delta}|x_i|$, which is equal to $\sum_{s_i=1} b_i^{\Delta}|x_i| + \sum_{t_i=1} b_i^{\Delta}|x_i|$, the second sum in (4.7). To prove the similar relations for the corresponding first sums let us note that diagonal terms (i.e., when $i = j$) in $Ineq(p,q)$ are nonnegative, while diagonal terms are zero in (4.7). We gather the remaining terms into symmetric pairs and show that for each $i < j$ one has

$$\begin{aligned} a_{ij}^{\Delta}|x_i x_j (p_i - q_j)| + a_{ij}^{\Delta}|x_j x_i (p_j - q_i)| &\geq \frac{1}{2}\Big(a_{ij}^{\Delta}|x_i x_j (s_i - s_j)| + a_{ij}^{\Delta}|x_i x_j (t_i - t_j)|\Big) \\ &\quad + \frac{1}{2}\Big(a_{ij}^{\Delta}|x_j x_i (s_j - s_i)| + a_{ij}^{\Delta}|x_j x_i (t_j - t_i)|\Big) \\ &= a_{ij}^{\Delta}|x_i x_j (s_i - s_j)| + a_{ij}^{\Delta}|x_i x_j (t_i - t_j)|. \end{aligned}$$

In fact, we prove a stronger inequality

$$|p_i - q_j| + |p_j - q_i| \geq |s_i - s_j| + |t_i - t_j|.$$

This can be shown simply by the enumeration of all possible values of p_i, p_j, q_i, and q_j, which is done in the following:

p_i	p_j	q_i	q_j	$\lvert p_i - q_j\rvert + \lvert p_j - q_i\rvert$	s_i	s_j	t_i	t_j	$\lvert s_i - s_j\rvert + \lvert t_i - t_j\rvert$
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1	0	1
0	1	0	0	1	0	0	0	1	1
1	1	0	0	2	0	0	1	1	0
0	0	1	0	1	0	0	1	0	1
1	0	1	0	2	1	0	1	0	2
0	1	1	0	0	0	0	1	1	0
1	1	1	0	1	1	0	1	1	1
0	0	0	1	1	0	0	0	1	1
1	0	0	1	0	0	0	1	1	0
0	1	0	1	2	0	1	0	1	2
1	1	0	1	1	0	1	1	1	1
0	0	1	1	2	0	0	1	1	0
1	0	1	1	1	1	0	1	1	1
0	1	1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1	1	0

Now we have that the right-hand side of $Ineq(p,q)$ is less or equal to the right-hand side of (4.7), and the left-hand side of $Ineq(p,q)$ is greater or equal to the left-hand side of (4.7). Therefore, $Ineq(p,q)$ is redundant, and (4.2) can be replaced by the system

$$\sum_{i<j} a_{ij}^{\Delta}|x_i x_j(p_i - p_j)| + \sum_{i=1}^{n} b_i^{\Delta}|x_i p_i| \geq \left|\sum_{i=1}^{n} r_i x_i p_i\right| \quad \forall\, p \in \{0,1\}^n \setminus \{0\}. \tag{4.8}$$

The last reduction follows from the fact that for each unit vector $p \equiv e_k$ the corresponding inequality in (4.8) represents an $|x_k|$-multiple of the kth Oettli–Prager inequality (4.5). □

The resulting number of inequalities in the description is again exponential. But in comparison with the upper bound $8\left(\frac{3}{2}\right)^{2^{\kappa+1}}$, $\kappa = \frac{1}{2}n(n+1)$, for the final number of inequalities obtained by Fourier–Motzkin elimination (see [4]), the improvement is significant. For $n = 2$, system (4.6) comprises one inequality, and for $n = 3$ we get four inequalities. In these cases, Fourier–Motzkin elimination yields two and eight inequalities, respectively.

Example 4.3. For $n = 2$, system (4.6) is composed of only one inequality

$$b_1^{\Delta}|x_1| + b_2^{\Delta}|x_2| \geq |b_1^c x_1 + b_2^c x_2|.$$

In this two-dimensional case the set Σ_{skew} represents a polyhedral set, which is convex in each orthant (cf. [4]). The following are some particular examples:

1. For $\boldsymbol{A} = \begin{pmatrix} 0 & [1,2] \\ [-2,-1] & 0 \end{pmatrix}$, $\boldsymbol{b} = \begin{pmatrix} [0,2] \\ [-2,2] \end{pmatrix}$ we have $\Sigma = \Sigma_{skew}$ and both are bounded.
2. For $\boldsymbol{A} = \begin{pmatrix} 0 & [-1,1] \\ [-1,1] & 0 \end{pmatrix}$, $\boldsymbol{b} = \begin{pmatrix} [0,2] \\ [-2,2] \end{pmatrix}$ we have $\Sigma = \Sigma_{skew}$ and both are unbounded.

REFERENCES

[1] G. Alefeld and J. Herzberger, *Introduction to Interval Computations*, Academic Press, London, 1983.

[2] G. Alefeld and G. Mayer, *On the symmetric and unsymmetric solution set of interval systems*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1223–1240.

[3] G. Alefeld, V. Kreinovich, and G. Mayer, *The shape of the symmetric solution set*, in Proceedings of the International Workshop on Applications of Interval Computations, El Paso, 1995, B. Kearfott and V. Kreinovich, eds., Kluwer Academic Publishers, Dordrecht, 1996.

[4] G. Alefeld, V. Kreinovich, and G. Mayer, *On the shape of the symmetric, persymmetric, and skew-symmetric solution set*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 693–705.

[5] G. Alefeld, V. Kreinovich, and G. Mayer, *The shape of the solution set for systems of interval linear equations with dependent coefficients*, Math. Nachr., 192 (1998), pp. 23–36.

[6] G. Alefeld, V. Kreinovich, and G. Mayer, *On symmetric solution sets*, in Inclusion Methods for Nonlinear Problems: With Applications in Engineering, Economics and Physics. Proceedings of the International GAMM-Workshop, Munich and Oberschleissheim, 2000, Comput. Suppl. 16, J. Herzberger, ed., Springer, Wien, 2003, pp. 1–22.

[7] G. Alefeld, V. Kreinovich, and G. Mayer, *On the solution sets of particular classes of linear interval systems*, J. Comput. Appl. Math., 152 (2003), pp. 1–15.

[8] R. Araiza, G. Xiang, O. Kosheleva, and D. Škulj, *Under interval and fuzzy uncertainty, symmetric Markov chains are more difficult to predict*, in Proceedings of the 26th International Conference of the North American Fuzzy Information Processing Society NAFIPS'2007, M. Reformat and M. R. Berthold, eds., San Diego, CA, 2007, pp. 526–531.

[9] M. Hladík, *Solution set characterization of linear interval systems with a specific dependence structure*, Reliab. Comput., 13 (2007), pp. 361–374.

[10] Z. Kulpa, A. Pownuk, and I. Skalna, *Analysis of linear mechanical structures with uncertainties by means of interval methods*, Comput. Assist. Mech. Eng. Sci., 5 (1998), pp. 443–477.

[11] W. Oettli and W. Prager, *Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides*, Numer. Math., 6 (1964), pp. 405–409.

[12] M. Padberg, *Linear Optimization and Extension*, Springer, Berlin, 1999.

[13] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons Ltd., Chichester, UK, 1998.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 522–537

ON CONVERGENCE OF THE DQDS ALGORITHM FOR SINGULAR VALUE COMPUTATION*

KENSUKE AISHIMA†, TAKAYASU MATSUO†, KAZUO MUROTA†, AND MASAAKI SUGIHARA†

Abstract. We prove global convergence, in exact arithmetic, for the differential quotient difference algorithm that is currently implemented in LAPACK for the computation of the singular values of a bidiagonal matrix. Our results cover any shift strategy that preserves positivity. We also show that the asymptotic rate for the Johnson shift is 3/2.

Key words. singular value, bidiagonal matrix, dqds algorithm, Johnson bound

AMS subject classifications. 15A18, 15A23, 65F20

DOI. 10.1137/060678762

1. Introduction. Every $n \times m$ real matrix A with $\mathrm{rank}(A) = r$ can be decomposed into

$$A = U\Sigma V^{\mathrm{T}}$$

with suitable orthogonal matrices $U \in \mathbf{R}^{n\times n}$ and $V \in \mathbf{R}^{m\times m}$, where

$$\Sigma = \begin{pmatrix} D & O_{r,m-r} \\ O_{n-r,r} & O_{n-r,m-r} \end{pmatrix}, \qquad D = \mathrm{diag}(\sigma_1, \ldots, \sigma_r),$$

and $\sigma_1 \geq \cdots \geq \sigma_r > 0$. The notation $O_{k,l}$ means a $k \times l$ zero matrix. The nonzero diagonal elements $\sigma_1, \ldots, \sigma_r$ are the nonzero singular values of A, which play important roles in application areas. Accordingly, numerical methods for computing singular values are of great importance in practice.

The singular values of A are equal to the square roots of the eigenvalues of $A^{\mathrm{T}}A$ and hence an iterative computation is inevitable for singular values. Usually, the given matrix A is first transformed to a bidiagonal matrix to reduce the overall computational cost. In the case of $n \geq m$, for example, the matrix A can be transformed, with appropriate orthogonal matrices $\tilde{U} \in \mathbf{R}^{n\times n}$ and $\tilde{V} \in \mathbf{R}^{m\times m}$, as

$$\tilde{U}^{\mathrm{T}} A \tilde{V} = \begin{pmatrix} B \\ O_{n-m,m} \end{pmatrix},$$

where $B \in \mathbf{R}^{m\times m}$ is an upper bidiagonal matrix. The singular values of B coincide with those of A.

Most of the current methods for computing singular values of bidiagonal matrices are based on the QR algorithm [3]. Demmel and Kahan's improvement [2] upon the

*Received by the editors December 28, 2006; accepted for publication (in revised form) by I. S. Dhillon December 26, 2007; published electronically May 16, 2008. Part of this work was supported by the 21st Century COE Program on Information Science and Technology Strategic Core and a Grant-in-Aid of the Ministry of Education, Culture, Sports, Science and Technology of Japan.
http://www.siam.org/journals/simax/30-2/67876.html

†Graduate School of Information Science and Technology, University of Tokyo, Tokyo, 113-8656, Japan (Kensuke_Aishima@mist.i.u-tokyo.ac.jp, matsuo@mist.i.u-tokyo.ac.jp, murota@mist.i.u-tokyo.ac.jp, m_sugihara@mist.i.u-tokyo.ac.jp).

QR algorithm, awarded the second SIAM prize in numerical linear algebra, is available as DBDSQR in LAPACK [1, 10]. The so-called differential quotient difference algorithm with shifts (dqds) was introduced by Fernando and Parlett in 1994 in [7]. The dqds algorithm has received majority support due to its accuracy, speed, and numerical stability, and is implemented as DLASQ in LAPACK. The dqds is integrated into multiple relatively robust representations (MR^3) algorithm [4, 5, 6]. The dqds algorithm that is implemented in subroutine DLASQ, and is much faster than the QR based DBDSQR, has an unusually complicated shift strategy. This strategy evolved in order to achieve efficiency both in the average case and in the worst case.

Although [7] describes various shift strategies it does not recommend any particular one, and that aspect is taken up in [13]. Consequently, [7] makes no mention of global convergence and might leave the impression that this property is in doubt. As we explain later, in this section, that fear is not warranted but the arguments are indirect and not readily accessible.

Our objective in this paper is to provide an elegant proof of global convergence of dqds in the context of exact arithmetic. The way we accomplish this task, without getting lost in the details of the strategy, is by first establishing global convergence for a sequence of nonnegative shifts that keeps the matrix entries positive (Theorem 4.1). Our analysis also enables us to give convergence rates for all of the matrix entries in terms of the accumulated shifts (Theorem 4.2). These are linear, as expected, for all of the entries except perhaps for the last row. When the accumulated shift converges to the smallest singular value squared, these last entries converge superlinearly. Furthermore we show that, if the shift is determined by the Johnson bound [9], then the asymptotic rate of convergence is 1.5 (Theorem 5.6).

Finally, we explain the present status of the convergence proof in some detail, primarily for the experts. In exact arithmetics, the dqds algorithm is equivalent to the Cholesky LR method with shifts for the eigenvalue computation of tridiagonal matrices. A convergence theorem for the latter is given by Rutishauser [15, Satz 2]. It should be noted, however, that the theorem involves a technical assumption that the case of "disorder of latent roots" is excluded. This means that the theorem applies in the (generic) case where the Cholesky LR method without shifts does not result in "disorder of latent roots." It is important to recognize that the theorem of Rutishauser [15], as it stands, is a generic convergence theorem which does not claim convergence in the exceptional case of "disorder of latent roots." In fact, this point is fully recognized by Rutishauser [14, 16] with a concrete example:

$$A = \begin{pmatrix} 5 & 4 & 1 & 1 \\ 4 & 5 & 1 & 1 \\ 1 & 1 & 4 & 2 \\ 1 & 1 & 2 & 4 \end{pmatrix}.$$

This matrix has eigenvalues 10, 5, 2, and 1, whereas the Cholesky LR method (without shifts) for this matrix converges to $\mathrm{diag}(10, 1, 5, 2)$. This example demonstrates that "disorder of latent roots" does occur, and, accordingly, Rutishauser's convergence theorem is indeed a generic convergence theorem.

For tridiagonal symmetric matrices, on the other hand, it is known in the literature (e.g., [7]) that the Cholesky LR method without shifts converges to a diagonal matrix with well-ordered diagonal elements. By combining this with Rutishauser's theorem mentioned above we can see that the Cholesky LR method with shifts is guaranteed to converge for tridiagonal symmetric matrices. This implies, in turn, that

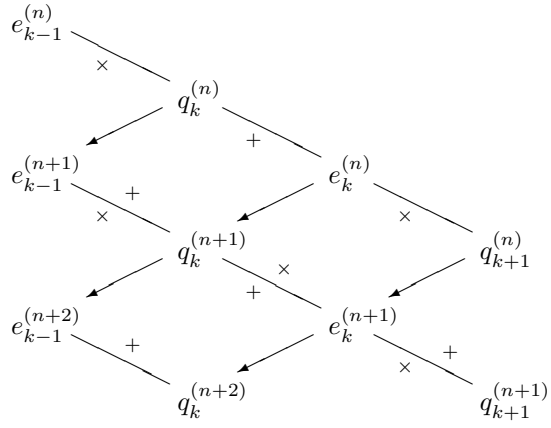


FIG. 1. *The rhombus rules.*

the dqds method is guaranteed to converge. Thus, a convergence proof of the dqds method for bidiagonal matrices can be obtained from a combination of known facts. Compared with this proof, our proof is simple, direct, and self-contained. Moreover, our proof is flexible enough to be modified to establish theorems about convergence rates, such as Theorem 5.6 for the Johnson shift.

2. Notation. Assume that the given real matrix A has already been transformed to a bidiagonal matrix

$$B = \begin{pmatrix} b_1 & b_2 & & \\ & b_3 & \ddots & \\ & & \ddots & b_{2m-2} \\ & & & b_{2m-1} \end{pmatrix}. \tag{2.1}$$

Following [7], we assume that the matrix is normalized to satisfy the following assumption.

Assumption (A). The bidiagonal elements of B are positive, i.e., $b_k > 0$ for $k = 1, 2, \ldots, 2m-1$.This assumption guarantees (see [12]) that the singular values of B are all distinct: $\sigma_1 > \cdots > \sigma_m > 0$.

In our problem setting we have assumed real matrices, whereas the singular value decomposition is also defined for complex matrices. Our restriction to real matrices is justified by the fact that any complex matrix can be transformed to a real bidiagonal matrix by, say, (complex) Householder transformations, while keeping its singular values [7].

3. The dqds algorithm. In this section, the dqds and related algorithms are summarized. Before describing the dqds algorithm, we review the pqds algorithm, which is mathematically equivalent to the dqds and serves as the main target in the subsequent theoretical analysis. The pqds algorithm is the pqd algorithm where shifts are incorporated to accelerate the convergence [8, 11, 16]. The pqd algorithm consists of the so-called [illegible] (Figure 1).

The pqds algorithm, in computer program form, is shown in Algorithm 3.1. The outermost loop is terminated when some suitable convergence criterion, say $|e_{m-1}^{(n)}| \leq \epsilon$

Algorithm 3.1 The pqds algorithm.

Initialization: $q_k^{(0)} = (b_{2k-1})^2$ $(k = 1, 2, \ldots, m)$; $e_k^{(0)} = (b_{2k})^2$ $(k = 1, 2, \ldots, m-1)$
1: **for** $n := 0, 1, \ldots$ **do**
2: choose shift $s^{(n)} (\geq 0)$
3: $e_0^{(n+1)} := 0$
4: **for** $k := 1, \ldots, m-1$ **do**
5: $q_k^{(n+1)} := q_k^{(n)} - e_{k-1}^{(n+1)} + e_k^{(n)} - s^{(n)}$
6: $e_k^{(n+1)} := e_k^{(n)} q_{k+1}^{(n)} / q_k^{(n+1)}$
7: **end for**
8: $q_m^{(n+1)} := q_m^{(n)} - e_{m-1}^{(n+1)} - s^{(n)}$
9: **end for**

for some prescribed constant $\epsilon > 0$, is satisfied. At the termination we have

$$\sigma_m{}^2 \approx q_m^{(n)} + \sum_{l=0}^{n-1} s^{(l)}, \tag{3.1}$$

and hence σ_m can be approximated by $\sqrt{q_m^{(n)} + \sum_{l=0}^{n-1} s^{(l)}}$. Then by the deflation process the problem is shrunk to an $(m-1)\times(m-1)$ problem, and the same procedure is repeated until $\sigma_{m-1}, \ldots, \sigma_1$ are obtained in turn.

It is convenient to introduce additional subdiagonal elements:

$$e_0^{(n)} = 0, \quad e_m^{(n)} = 0 \quad (n = 0, 1, \ldots) \tag{3.2}$$

to simplify the expression of the algorithm. Put

$$B^{(n)} = \begin{pmatrix} b_1^{(n)} & b_2^{(n)} & & \\ & b_3^{(n)} & \ddots & \\ & & \ddots & b_{2m-2}^{(n)} \\ & & & b_{2m-1}^{(n)} \end{pmatrix}, \tag{3.3}$$

$b_k^{(0)} = b_k$ $(k = 1, 2, \ldots, 2m-1)$, and

$$q_k^{(n)} = (b_{2k-1}^{(n)})^2 \quad (k = 1, 2, \ldots, m;\ n = 0, 1, \ldots), \tag{3.4}$$

$$e_k^{(n)} = (b_{2k}^{(n)})^2 \quad (k = 1, 2, \ldots, m-1;\ n = 0, 1, \ldots). \tag{3.5}$$

Then Algorithm 3.1 can be rewritten in terms of the Cholesky decomposition (with shifts):

$$(B^{(n+1)})^{\mathrm{T}} B^{(n+1)} = B^{(n)} (B^{(n)})^{\mathrm{T}} - s^{(n)} I, \tag{3.6}$$

where $B^{(0)} = B$. It follows that

$$(B^{(n)})^{\mathrm{T}} B^{(n)} = W^{(n)} \left((B^{(0)})^{\mathrm{T}} B^{(0)} - \sum_{l=0}^{n-1} s^{(l)} I \right) (W^{(n)})^{-1}, \tag{3.7}$$

where $W^{(n)} = (B^{(n-1)} \cdots B^{(0)})^{-\mathrm{T}}$ is a nonsingular matrix (see Lemma 3.1). Therefore the eigenvalues of $(B^{(n)})^{\mathrm{T}} B^{(n)}$ are the same as those of $(B^{(0)})^{\mathrm{T}} B^{(0)} - \sum_{l=0}^{n-1} s^{(l)} I$. In

actual computation it is often observed that $B^{(n)}$ converges to a diagonal matrix as $n \to \infty$, and then, by (3.7), the singular values of B can be obtained from the diagonal elements of $B^{(n)}$ for sufficiently large n. We give a theoretical proof for the global convergence in the next section.

The following lemma states that, if $s^{(n)} < (\sigma_{\min}^{(n)})^2$ in each iteration n, where $\sigma_{\min}^{(n)}$ is the smallest singular value of $B^{(n)}$, then the variables in the pqds algorithm are always positive so that the algorithm does not break down.

LEMMA 3.1 (positivity of the variables in the pqds algorithm). [illegible] B [illegible] (A) [illegible] $s^{(n)} < (\sigma_{\min}^{(n)})^2$ $(n = 0, 1, 2, \ldots)$ [illegible] $(B^{(n)})^{\mathrm{T}} B^{(n)}$ [illegible] $q_k^{(n)} > 0$ $(k = 1, \ldots, m)$ [illegible] $e_k^{(n)} > 0$ $(k = 1, \ldots, m-1)$ [illegible] $n = 0, 1, 2, \ldots$

[illegible]. We prove by induction. Under Assumption (A), we have $q_k^{(0)} > 0$, $e_k^{(0)} > 0$, and that $(B^{(0)})^{\mathrm{T}} B^{(0)}$ is positive definite. Suppose that $(B^{(n)})^{\mathrm{T}} B^{(n)}$ is positive definite and $q_k^{(n)} > 0$, $e_k^{(n)} > 0$. By (3.6), if $s^{(n)} < (\sigma_{\min}^{(n)})^2$, then $(B^{(n+1)})^{\mathrm{T}} B^{(n+1)}$ is positive definite because $B^{(n)}(B^{(n)})^{\mathrm{T}} - s^{(n)} I$ is positive definite. Therefore all of the diagonal elements of B are nonzero ($b_{2k-1}^{(n+1)} \neq 0$), and hence $q_k^{(n+1)} > 0$ because of (3.4). By line 6 of Algorithm 3.1, we have $e_k^{(n+1)} > 0$. □

The dqds algorithm is obtained from the pqds algorithm by introducing the auxiliary quantities $d_k^{(n+1)}$ defined as follows [7]:

$$d_1^{(n+1)} = q_1^{(n)} - s^{(n)}; \quad d_k^{(n+1)} = q_k^{(n)} - e_{k-1}^{(n+1)} - s^{(n)} \quad (k = 2, \ldots, m). \tag{3.8}$$

The resulting algorithm is presented as Algorithm 3.2. Generally, the dqds algorithm is more accurate than the pqds algorithm. Since the variables of the dqds algorithm are positive (see Lemma 3.2) and no subtractions are used in the algorithm except for computing the shifts, the numerical instability due to a loss of significant digits is less likely to happen in the dqds algorithm.

Algorithm 3.2 The dqds algorithm.

Initialization: $q_k^{(0)} = (b_{2k-1})^2$ $(k = 1, 2, \ldots, m)$; $e_k^{(0)} = (b_{2k})^2$ $(k = 1, 2, \ldots, m-1)$

1: **for** $n := 0, 1, \ldots$ **do**
2: choose shift $s^{(n)} (\geq 0)$
3: $d_1^{(n+1)} := q_1^{(n)} - s^{(n)}$
4: **for** $k := 1, \ldots, m-1$ **do**
5: $q_k^{(n+1)} := d_k^{(n+1)} + e_k^{(n)}$
6: $e_k^{(n+1)} := e_k^{(n)} q_{k+1}^{(n)} / q_k^{(n+1)}$
7: $d_{k+1}^{(n+1)} := d_k^{(n+1)} q_{k+1}^{(n)} / q_k^{(n+1)} - s^{(n)}$
8: **end for**
9: $q_m^{(n+1)} := d_m^{(n+1)}$
10: **end for**

LEMMA 3.2 (positivity of the variables in the dqds algorithm). [illegible] B [illegible] (A) [illegible] $s^{(n)} < (\sigma_{\min}^{(n)})^2$ $(n = 0, 1, 2, \ldots)$ [illegible] $(B^{(n)})^{\mathrm{T}} B^{(n)}$ [illegible] $q_k^{(n)} > 0$ $(k = 1, \ldots, m)$, $e_k^{(n)} > 0$ $(k = 1, \ldots, m-1)$ [illegible] $d_k^{(n)} > 0$ $(k = 1, \ldots, m)$ [illegible] $n = 0, 1, 2, \ldots$

[illegible]. By Lemma 3.1, we have $e_k^{(n)} > 0$ and $q_k^{(n)} > 0$. The inequality $d_k^{(n)} > 0$ is proved by contradiction as follows. If we had $d_k^{(n)} \leq 0$ for some k, then we would

have $d_{k+1}^{(n)} \leq 0$ by line 7 of Algorithm 3.2, and then $q_m^{(n)} = d_m^{(n)} \leq 0$. This contradicts $q_m^{(n)} > 0$. □

4. Convergence of the dqds. In this section,we prove that, for any matrix B that satisfies Assumption (A), the variables $q_k^{(n)}$ and $e_k^{(n)}$ in the dqds algorithm converge as far as the shift is chosen such that $0 \leq s^{(n)} < (\sigma_{\min}^{(n)})^2$, where $\sigma_{\min}^{(n)}$ is the smallest singular value of $B^{(n)}$.

The next theorem establishes the convergence of the dqds. Moreover, the theorem states that the variables $q_k^{(n)}$ converge to the square of the singular values minus the sum of the shifts, and that they are placed in the descending order.

THEOREM 4.1 (convergence of the dqds algorithm). [illegible] B [illegible] (A) [illegible] $0 \leq s^{(n)} < (\sigma_{\min}^{(n)})^2$ [illegible]

$$\sum_{n=0}^{\infty} s^{(n)} \leq {\sigma_m}^2. \tag{4.1}$$

[illegible]

$$\lim_{n\to\infty} e_k^{(n)} = 0 \quad (k = 1, 2, \ldots, m-1), \tag{4.2}$$

$$\lim_{n\to\infty} q_k^{(n)} = {\sigma_k}^2 - \sum_{n=0}^{\infty} s^{(n)} \quad (k = 1, 2, \ldots, m). \tag{4.3}$$

[illegible]

$$\lim_{n\to\infty} (B^{(n)})^{\mathrm{T}} B^{(n)} = \operatorname{diag}\left({\sigma_1}^2 - \sum_{n=0}^{\infty} s^{(n)}, \ldots, {\sigma_m}^2 - \sum_{n=0}^{\infty} s^{(n)} \right).$$

[illegible]. On the basis of the equivalence between the dqds algorithm and the pqds algorithm, we show the convergence of the pqds to prove this theorem.

By the assumption and Lemma 3.1, $(B^{(n)})^{\mathrm{T}} B^{(n)}$ is a positive-definite symmetric matrix. It then follows from (3.7) that

$$\sum_{n=0}^{N} s^{(n)} < {\sigma_m}^2 \tag{4.4}$$

holds for any $N \geq 1$. In the limit of $N \to \infty$, we obtain (4.1).

Next, we prove $\lim_{n\to\infty} e_k^{(n)} = 0$. By Lemma 3.1, we have $e_k^{(n)} > 0$. Therefore it is sufficient to prove $\sum_{n=0}^{\infty} e_k^{(n)} < +\infty$. Adding both sides of line 5 of Algorithm 3.1 for over n with k fixed, we obtain

$$q_k^{(n+1)} = q_k^{(0)} + \sum_{l=0}^{n} e_k^{(l)} - \sum_{l=0}^{n} e_{k-1}^{(l+1)} - \sum_{l=0}^{n} s^{(l)} \quad (k = 1, 2, \ldots, m). \tag{4.5}$$

Since $q_k^{(n+1)} > 0$ by Lemma 3.1, it follows that

$$\sum_{l=0}^{n} e_{k-1}^{(l+1)} < q_k^{(0)} + \sum_{l=0}^{n} e_k^{(l)} - \sum_{l=0}^{n} s^{(l)} \leq q_k^{(0)} + \sum_{l=0}^{n} e_k^{(l)} \quad (k = 1, 2, \ldots, m). \tag{4.6}$$

Setting $k = m$ in (4.6), we obtain $\sum_{l=0}^{\infty} e_{m-1}^{(l+1)} \leq q_m^{(0)}$, with the aid of (3.2). Similarly, setting $k = m-1,\ m-2, \ldots,\ 2$ in (4.6), we obtain

$$\sum_{l=0}^{\infty} e_k^{(l+1)} \leq \sum_{j=k+1}^{m} q_j^{(0)} < +\infty \qquad (k = m-1, m-2, \ldots, 1),$$

which completes the proof for $e_k^{(n)}$.

Next, we prove (4.3). By (4.5) with $n \to \infty$, we see

$$\lim_{n\to\infty} q_k^{(n)} = q_k^{(0)} + \lim_{n\to\infty} \sum_{l=0}^{n} e_k^{(l)} - \lim_{n\to\infty} \sum_{l=0}^{n} e_{k-1}^{(l+1)} - \lim_{n\to\infty} \sum_{l=0}^{n} s^{(l)}. \tag{4.7}$$

Since the right-hand side of (4.7) converges, $q_k^{(\infty)} = \lim_{n\to\infty} q_k^{(n)}$ exists. Because $\lim_{n\to\infty} e_k^{(n)} = 0$, (3.7) reads

$$\begin{aligned} &\lim_{n\to\infty} W^{(n)} \left((B^{(0)})^{\mathrm{T}} B^{(0)} - \sum_{l=0}^{n-1} s^{(l)} I \right) (W^{(n)})^{-1} \\ &= \lim_{n\to\infty} (B^{(n)})^{\mathrm{T}} B^{(n)} = \mathrm{diag}(q_1^{(\infty)}, \ldots, q_m^{(\infty)}), \end{aligned}$$

which shows the convergence of the form

$$q_k^{(\infty)} = {\sigma_{p(k)}}^2 - \sum_{l=0}^{\infty} s^{(l)} \qquad (k = 1, \ldots, m),$$

where $p(k)$ denotes a permutation of indices k $(k = 1, \ldots, m)$. It remains to show that $q_k^{(\infty)}$ are in the descending order. From line 6 of Algorithm 3.1, we have

$$e_k^{(n)} = e_k^{(0)} \prod_{l=0}^{n-1} \frac{q_{k+1}^{(l)}}{q_k^{(l+1)}} \quad (k = 1, \ldots, m-1).$$

Because all of the singular values are distinct, $\sigma_1 > \cdots > \sigma_m$, by the assumption, the limits $q_1^{(\infty)}, \ldots, q_m^{(\infty)}$ are also distinct. Since $\lim_{n\to\infty} e_k^{(n)} = 0$, we have

$$q_k^{(\infty)} > q_{k+1}^{(\infty)} \quad (k = 1, 2, \ldots, m-1).$$

This completes the proof of Theorem 4.1. □

The next theorem states the asymptotic rate of convergence of the dqds algorithm. Let us define

$$\rho_k = \frac{{\sigma_{k+1}}^2 - \sum_{n=0}^{\infty} s^{(n)}}{{\sigma_k}^2 - \sum_{n=0}^{\infty} s^{(n)}} \qquad (k = 1, \ldots, m-1), \tag{4.8}$$

$$r_k^{(n)} = \left(q_k^{(n)} + \sum_{l=0}^{n-1} s^{(l)} \right) - {\sigma_k}^2 \qquad (k = 1, \ldots, m). \tag{4.9}$$

In view of (3.1), $r_k^{(n)}$ is the error in the approximated eigenvalue of $B^{\mathrm{T}}B$. Note that $0 < \rho_k < 1$ $(k = 1, \ldots, m-2)$, and $0 < \rho_{m-1} < 1$ if ${\sigma_m}^2 - \sum_{n=0}^{\infty} s^{(n)} > 0$ and $\rho_{m-1} = 0$ if ${\sigma_m}^2 - \sum_{n=0}^{\infty} s^{(n)} = 0$.

THEOREM 4.2 (rate of convergence of the dqds algorithm). [illegible] 4.1 [illegible]

$$\lim_{n\to\infty} \frac{e_k^{(n+1)}}{e_k^{(n)}} = \rho_k \qquad (k = 1, \ldots, m-1), \tag{4.10}$$

$$\lim_{n\to\infty} \frac{r_1^{(n+1)}}{r_1^{(n)}} = \rho_1, \tag{4.11}$$

$$\lim_{n\to\infty} \frac{r_m^{(n+1)}}{r_m^{(n)}} = \rho_{m-1}. \tag{4.12}$$

[illegible] $\rho_{k-1} \neq \rho_k$ $(k = 2, \ldots, m-1)$ [illegible]

$$\lim_{n\to\infty} \frac{r_k^{(n+1)}}{r_k^{(n)}} = \max\{\rho_{k-1}, \rho_k\} \qquad (k = 2, \ldots, m-1). \tag{4.13}$$

[illegible] $e_k^{(n)}$ $(k = 1, \ldots, m-2)$ [illegible] $r_k^{(n)}$ $(k = 1, \ldots, m-1)$ [illegible] $n \to \infty$ [illegible] $e_{m-1}^{(n)}$ [illegible] $r_m^{(n)}$ [illegible] $\rho_{m-1} > 0$, [illegible] $\sigma_m^2 - \sum_{n=0}^{\infty} s^{(n)} > 0$, [illegible] $\rho_{m-1} = 0$, [illegible] $\sigma_m^2 - \sum_{n=0}^{\infty} s^{(n)} = 0$ [illegible]. From line 6 of Algorithm 3.1, we have

$$\frac{e_k^{(n+1)}}{e_k^{(n)}} = \frac{q_{k+1}^{(n)}}{q_k^{(n+1)}} \quad (k = 1, \ldots, m-1).$$

Then (4.10) is obvious from (4.3) and (4.8).

In order to prove the rest of the theorem, we first express $r_k^{(n)}$ in terms of $e_k^{(n)}$ whose asymptotic behavior is now known. From (4.5), we have

$$q_k^{(n)} + \sum_{l=0}^{n-1} s^{(l)} = q_k^{(0)} + \sum_{l=0}^{n-1} e_k^{(l)} - \sum_{l=0}^{n-1} e_{k-1}^{(l+1)}. \tag{4.14}$$

From (4.3) and (4.7), we also have

$$\sigma_k{}^2 = q_k^{(\infty)} + \sum_{l=0}^{\infty} s^{(l)} = q_k^{(0)} + \sum_{l=0}^{\infty} e_k^{(l)} - \sum_{l=0}^{\infty} e_{k-1}^{(l+1)}. \tag{4.15}$$

Subtracting (4.15) from (4.14), we obtain

$$r_k^{(n)} = \left(q_k^{(n)} + \sum_{l=0}^{n-1} s^{(l)} \right) - \sigma_k{}^2 = \sum_{l=n}^{\infty} e_{k-1}^{(l+1)} - \sum_{l=n}^{\infty} e_k^{(l)} \quad (k = 1, \ldots, m). \tag{4.16}$$

Thus, our task is now to evaluate

$$\frac{r_k^{(n+1)}}{r_k^{(n)}} = \frac{\sum_{l=n+1}^{\infty} e_{k-1}^{(l+1)} - \sum_{l=n+1}^{\infty} e_k^{(l)}}{\sum_{l=n}^{\infty} e_{k-1}^{(l+1)} - \sum_{l=n}^{\infty} e_k^{(l)}} \tag{4.17}$$

for $k = 1, \dots, m$ in the limit of $n \to \infty$. When $k = 1$, $e_0^{(l)}$ vanishes due to (3.2), and thus

$$\lim_{n\to\infty} \frac{r_1^{(n+1)}}{r_1^{(n)}} = \lim_{n\to\infty} \left(\frac{e_1^{(n)}}{\sum_{l=n}^{\infty} e_1^{(l)}} \cdot \frac{\sum_{l=n+1}^{\infty} e_1^{(l)}}{e_1^{(n+1)}} \cdot \frac{e_1^{(n+1)}}{e_1^{(n)}} \right) = \rho_1, \tag{4.18}$$

which is the claim (4.10). In the calculation above, we used the identity

$$\lim_{n\to\infty} \frac{\sum_{l=n}^{\infty} e_k^{(l)}}{e_k^{(n)}} = \sum_{l=0}^{\infty} \lim_{n\to\infty} \left(\frac{e_k^{(n+l)}}{e_k^{(n)}} \right) = \sum_{l=0}^{\infty} (\rho_k)^l = \frac{1}{1-\rho_k} \quad (k = 1, \dots, m-1), \tag{4.19}$$

which is obtained from (4.10). The claim (4.12) can be proved in a similar manner.

In the cases where $k = 2, \dots, m-1$, the assumption $\rho_{k-1} \neq \rho_k$ $(k = 2, \dots, m-1)$ is required. First, let us fix some k and consider the case $\rho_{k-1} > \rho_k$. In this case, an identity

$$\lim_{n\to\infty} \frac{e_k^{(n)}}{e_{k-1}^{(n)}} = 0 \tag{4.20}$$

holds, because

$$\lim_{n\to\infty} \left\{ \left(\frac{e_k^{(n+1)}}{e_{k-1}^{(n+1)}} \right) \cdot \left(\frac{e_k^{(n)}}{e_{k-1}^{(n)}} \right)^{-1} \right\} = \lim_{n\to\infty} \left\{ \left(\frac{e_k^{(n+1)}}{e_k^{(n)}} \right) \cdot \left(\frac{e_{k-1}^{(n)}}{e_{k-1}^{(n+1)}} \right) \right\} = \frac{\rho_k}{\rho_{k-1}} < 1.$$

Therefore, from (4.19) and (4.20) we obtain

$$\lim_{n\to\infty} \frac{\sum_{l=n}^{\infty} e_k^{(l)}}{e_{k-1}^{(n)}} = \lim_{n\to\infty} \left(\frac{e_k^{(n)}}{e_{k-1}^{(n)}} \cdot \frac{\sum_{l=n}^{\infty} e_k^{(l)}}{e_k^{(n)}} \right) = 0, \tag{4.21}$$

which then yields

$$\begin{aligned} \lim_{n\to\infty} \frac{\sum_{l=n+1}^{\infty} e_{k-1}^{(l+1)} - \sum_{l=n+1}^{\infty} e_k^{(l)}}{e_{k-1}^{(n+1)}} &= \lim_{n\to\infty} \left(\frac{\sum_{l=n+1}^{\infty} e_{k-1}^{(l)}}{e_{k-1}^{(n+1)}} - 1 - \frac{\sum_{l=n+1}^{\infty} e_k^{(l)}}{e_{k-1}^{(n+1)}} \right) \\ &= \frac{1}{1-\rho_{k-1}} - 1 \\ &= \frac{\rho_{k-1}}{1-\rho_{k-1}} \end{aligned}$$

with (4.19). Finally, we obtain

$$\begin{aligned} \lim_{n\to\infty} \frac{r_k^{(n+1)}}{r_k^{(n)}} &= \lim_{n\to\infty} \frac{\sum_{l=n+1}^{\infty} e_{k-1}^{(l+1)} - \sum_{l=n+1}^{\infty} e_k^{(l)}}{\sum_{l=n}^{\infty} e_{k-1}^{(l+1)} - \sum_{l=n}^{\infty} e_k^{(l)}} \\ &= \lim_{n\to\infty} \left\{ \frac{(\sum_{l=n+1}^{\infty} e_{k-1}^{(l+1)} - \sum_{l=n+1}^{\infty} e_k^{(l)})/e_{k-1}^{(n+1)}}{(\sum_{l=n}^{\infty} e_{k-1}^{(l+1)} - \sum_{l=n}^{\infty} e_k^{(l)})/e_{k-1}^{(n)}} \cdot \frac{e_{k-1}^{(n+1)}}{e_{k-1}^{(n)}} \right\} \\ &= \rho_{k-1}. \end{aligned} \tag{4.22}$$

When $\rho_{k-1} < \rho_k$, a similar argument leads us to the same conclusion with ρ_{k-1} replaced by ρ_k. This completes the proof. □

. . . . 1. When $\rho_{k-1} = \rho_k$, considering the ratio (4.17) does not make sense, since the numerator and the denominator can oscillate around zero. Instead, we can show a weaker claim as follows. From (4.16) we have

$$r_k^{(n)} = \frac{\sum_{l=n+1}^{\infty} e_{k-1}^{(l)}}{e_{k-1}^{(n+1)}} \cdot e_{k-1}^{(n+1)} - \frac{\sum_{l=n}^{\infty} e_k^{(l)}}{e_k^{(n)}} \cdot e_k^{(n)}.$$

Moreover, from (4.10) we see, for any small $\epsilon > 0$,

$$|e_{k-1}^{(n)}| \le O((\rho_{k-1} + \epsilon)^n), \quad |e_k^{(n)}| \le O((\rho_k + \epsilon)^n).$$

Thus, in case of $\rho_{k-1} = \rho_k$, we obtain

$$|r_k^{(n)}| \le O((\rho_k + \epsilon)^n) \qquad (k = 2, \ldots, m-1)$$

from (4.19). That is, the convergence is at least linear and can be even better occasionally.

5. Convergence rate of the dqds with the Johnson bound. In this section, we prove that the asymptotic rate of convergence of the dqds algorithm is 1.5 if the shift is determined by the Johnson bound [9]. In the proofs we will work with the pqds in place of the dqds, as we did in the previous section.

Though the Johnson bound is valid for a general matrix, we present here its version for a bidiagonal matrix B.

LEMMA 5.1 (Johnson bound [9]). . . . . B . . . . . (2.1) . . .

$$\lambda = \min_{k=1,\ldots,m} \left\{ |\, b_{2k-1} \,| - \frac{|\, b_{2k-2} \,| + |\, b_{2k} \,|}{2} \right\},$$

. . $b_0 = b_{2m} = 0$ . . . . . σ_m . . . . . B . . . $\sigma_m \ge \lambda$ . . . . . $(b_2, b_4, \ldots, b_{2m-2})$ . . . . . $\sigma_m > \lambda$

With reference to (3.3), (3.4), and (3.5), we define the shift by the Johnson bound as follows:

$$\lambda^{(n)} = \min_{k=1,\ldots,m} \left\{ \sqrt{q_k^{(n)}} - \frac{1}{2} \left(\sqrt{e_{k-1}^{(n)}} + \sqrt{e_k^{(n)}} \right) \right\}, \tag{5.1}$$

$$s^{(n)} = \left(\max\{\lambda^{(n)},\ 0\} \right)^2. \tag{5.2}$$

This choice of the shift guarantees the condition $0 \le s^{(n)} < (\sigma_{\min}^{(n)})^2$ in each iteration n, and hence the dqds is convergent by Theorem 4.1. The precise rate of convergence can be revealed through a scrutiny of the shift.

The next lemma shows that the Johnson bound $\lambda^{(n)}$ is determined solely by $q_m^{(n)}$ and $e_{m-1}^{(n)}$ when n is large enough. As a corollary of this fact we see that $q_m^{(n)}$ approaches zero.

LEMMA 5.2. . . . . (A) . . . . . (5.2) . . . . . n . . . . .

$$\lambda^{(n)} = \sqrt{q_m^{(n)}} - \frac{1}{2}\sqrt{e_{m-1}^{(n)}}. \tag{5.3}$$

(5.1) $k = m$. Let $k < m$, and consider the identity

$$\left[\sqrt{q_k^{(n)}} - \frac{1}{2}\left(\sqrt{e_{k-1}^{(n)}} + \sqrt{e_k^{(n)}}\right)\right] - \left[\sqrt{q_m^{(n)}} - \frac{1}{2}\left(\sqrt{e_{m-1}^{(n)}} + \sqrt{e_m^{(n)}}\right)\right]$$
$$= \left(\sqrt{q_k^{(n)}} - \sqrt{q_m^{(n)}}\right) - \frac{1}{2}\left(\sqrt{e_{k-1}^{(n)}} + \sqrt{e_k^{(n)}} - \sqrt{e_{m-1}^{(n)}} - \sqrt{e_m^{(n)}}\right).$$

From Theorem 4.1, the first term on the right-hand side remains positive:

$$\lim_{n\to\infty}\left(\sqrt{q_k^{(n)}} - \sqrt{q_m^{(n)}}\right) = \sqrt{{\sigma_k}^2 - \sum_{n=0}^{\infty} s^{(n)}} - \sqrt{{\sigma_m}^2 - \sum_{n=0}^{\infty} s^{(n)}} > 0,$$

while the second term vanishes since $\lim_{n\to\infty} e_k^{(n)} = 0$ for each k. Thus the minimum on the right-hand side of (5.1) is attained at $k = m$. □

LEMMA 5.3. 5.2

$$\sum_{n=0}^{\infty} s^{(n)} = {\sigma_m}^2, \tag{5.4}$$

$$\lim_{n\to\infty} q_k^{(n)} = {\sigma_k}^2 - {\sigma_m}^2 \quad (k = 1, \ldots, m-1); \quad \lim_{n\to\infty} q_m^{(n)} = 0. \tag{5.5}$$

. By (5.3) and (4.2), $\lim_{n\to\infty} \lambda^{(n)} = \lim_{n\to\infty} \sqrt{q_m^{(n)}} \geq 0$, and hence

$$\lim_{n\to\infty} s^{(n)} = \lim_{n\to\infty} (\max\{\lambda^{(n)}, 0\})^2 = \lim_{n\to\infty} q_m^{(n)}.$$

Since $\lim_{n\to\infty} s^{(n)} = 0$ by (4.1), we have $\lim_{n\to\infty} q_m^{(n)} = 0$. This, together with (4.3), proves (5.4) and (5.5). □

The next lemma shows $\lambda^{(n)} > 0$ for all sufficiently large n.

LEMMA 5.4 (positivity of the Johnson bound in the dqds). 5.2 N $\lambda^{(n)} > 0$ $n > N$

. The proof consists of showing two facts: (i) For every integer N', there exists $n > N'$ such that $\lambda^{(n)} > 0$; (ii) There exists an integer N'' such that, for all $n > N''$, $\lambda^{(n)} > 0$ implies $\lambda^{(n+1)} > 0$.

(i) The proof is done by contradiction. Suppose that there exists some N' such that $\lambda^{(n)} \leq 0$ for every $n > N'$. Then $s^{(n)} = 0$ (for all $n > N'$), and by (4.3) and (4.4) in Theorem 4.1, we have

$$\lim_{n\to\infty} q_m^{(n)} = {\sigma_m}^2 - \sum_{n=0}^{\infty} s^{(n)} = {\sigma_m}^2 - \sum_{n=0}^{N'} s^{(n)} > 0,$$

which contradicts Lemma 5.3.

(ii) Assume $\lambda^{(n)} > 0$ for some large n such that (5.3) holds. In this case, $s^{(n)} =$

$(\lambda^{(n)})^2$ and

$$
\begin{aligned}
q_m^{(n+1)} &= q_m^{(n)} - e_{m-1}^{(n+1)} - s^{(n)} \\
&= \sqrt{e_{m-1}^{(n)} q_m^{(n)}} - e_{m-1}^{(n+1)} - \frac{1}{4} e_{m-1}^{(n)} \\
&> \frac{1}{2} \sqrt{e_{m-1}^{(n)} q_m^{(n)}} - e_{m-1}^{(n+1)} \\
&= \sqrt{e_{m-1}^{(n+1)}} \left(\frac{1}{2} \sqrt{q_{m-1}^{(n+1)}} - \sqrt{e_{m-1}^{(n+1)}} \right),
\end{aligned} \tag{5.6}
$$

where line 5 of Algorithm 3.1 is used in the first equality, (5.3) in the second equality, the assumption $\lambda^{(n)} > 0$ $\left(\text{i.e.,} \sqrt{q_m^{(n)}} > \frac{1}{2}\sqrt{e_{m-1}^{(n)}}\right)$ in the inequality, and line 6 of Algorithm 3.1 in the last equality. From (5.6) it follows that

$$
\sqrt{q_{m-1}^{(n+1)}} > \frac{5}{2} \sqrt{e_{m-1}^{(n+1)}} \implies q_m^{(n+1)} > \frac{1}{4} e_{m-1}^{(n+1)} \iff \lambda^{(n+1)} > 0.
$$

Since $\lim_{n\to\infty} q_{m-1}^{(n+1)} > 0$ and $\lim_{n\to\infty} e_{m-1}^{(n+1)} = 0$, there exists an integer N'' such that the first inequality holds for all $n > N''$. □

Using Lemmas 5.2 and 5.4, we see that for sufficiently large n the shift is given as follows.

LEMMA 5.5 (shift in the dqds). [illegible] 5.2 [illegible]

$$
s^{(n)} = (\lambda^{(n)})^2 = q_m^{(n)} - \sqrt{e_{m-1}^{(n)} q_m^{(n)}} + \frac{1}{4} e_{m-1}^{(n)} \; > 0 \tag{5.7}
$$

[illegible] n

We are now in position to prove that the rate of convergence of the dqds is 1.5. The next theorem refers only to the lower right two elements of $B^{(n)}$, and the error in the approximation of the smallest eigenvalue of $B^{\mathrm{T}}B$. This is sufficient from the practical point of view since whenever the lower right elements converge to zero, the deflation is applied to reduce the matrix size.

THEOREM 5.6 (rate of convergence of the dqds). [illegible] B [illegible] (A) [illegible]

$$
\lim_{n\to\infty} \frac{e_{m-1}^{(n+1)}}{(e_{m-1}^{(n)})^{3/2}} = \frac{1}{\sqrt{{\sigma_{m-1}}^2 - {\sigma_m}^2}}, \tag{5.8}
$$

$$
\lim_{n\to\infty} \frac{q_m^{(n+1)}}{(q_m^{(n)})^{3/2}} = \frac{1}{\sqrt{{\sigma_{m-1}}^2 - {\sigma_m}^2}}, \tag{5.9}
$$

$$
\lim_{n\to\infty} \frac{r_m^{(n+1)}}{(r_m^{(n)})^{3/2}} = \frac{1}{\sqrt{{\sigma_{m-1}}^2 - {\sigma_m}^2}}. \tag{5.10}
$$

[illegible] 1.5 [illegible] (3.3) [illegible] $B^{(n)}$ [illegible] $b_{2m-2}^{(n)}$ [illegible] $b_{2m-1}^{(n)}$ [illegible] 0 [illegible] 1.5 [illegible]

[illegible] $B^{\mathrm{T}}B$ [illegible] 0 [illegible]

$$\lim_{n\to\infty} \frac{\sqrt{e_{m-1}^{(n)}}}{q_m^{(n)}} = \frac{1}{\sqrt{{\sigma_{m-1}}^2 - {\sigma_m}^2}}, \tag{5.11}$$

$$\lim_{n\to\infty} \frac{r_m^{(n)}}{e_{m-1}^{(n)}} = 0. \tag{5.12}$$

[illegible] First, we compute the rate of convergence of $e_{m-1}^{(n)}$. By Lemma 5.5 the shift is determined by (5.7) for sufficiently large n, and we have

$$q_m^{(n+1)} = \sqrt{e_{m-1}^{(n)} q_m^{(n)}} - e_{m-1}^{(n+1)} - \frac{1}{4} e_{m-1}^{(n)} \tag{5.13}$$

from the second equality in (5.6). We also have

$$q_m^{(n+1)} = q_{m-1}^{(n+2)} e_{m-1}^{(n+2)} / e_{m-1}^{(n+1)}, \qquad q_m^{(n)} = q_{m-1}^{(n+1)} e_{m-1}^{(n+1)} / e_{m-1}^{(n)}, \tag{5.14}$$

from the 6th line of Algorithm 3.1. Thus we have

$$\begin{aligned} \frac{e_{m-1}^{(n+2)}}{(e_{m-1}^{(n+1)})^{3/2}} &= \frac{q_m^{(n+1)}}{q_{m-1}^{(n+2)}} \cdot \sqrt{\frac{q_{m-1}^{(n+1)}}{e_{m-1}^{(n)} q_m^{(n)}}} \\ &= \frac{\sqrt{q_{m-1}^{(n+1)}}}{q_{m-1}^{(n+2)}} \left(1 - \frac{e_{m-1}^{(n+1)}}{\sqrt{e_{m-1}^{(n)} q_m^{(n)}}} - \frac{e_{m-1}^{(n)}}{4\sqrt{e_{m-1}^{(n)} q_m^{(n)}}} \right) \\ &= \frac{\sqrt{q_{m-1}^{(n+1)}}}{q_{m-1}^{(n+2)}} \left(1 - \frac{\sqrt{e_{m-1}^{(n+1)}}}{\sqrt{q_{m-1}^{(n+1)}}} - \frac{1}{4\sqrt{q_{m-1}^{(n+1)}}} \cdot \frac{e_{m-1}^{(n)}}{\sqrt{e_{m-1}^{(n+1)}}} \right). \end{aligned} \tag{5.15}$$

The first equality is from (5.14), the second one is from (5.13), and the last one is from the second equation of (5.14). In the rest of this paragraph, we prove that the value in the parentheses on the right-hand side of (5.15) converges to 1. First, note that $\lim_{n\to\infty} q_{m-1}^{(n+1)} > 0$ by (5.5). By (4.2), $\lim_{n\to\infty} e_{m-1}^{(n+1)} = 0$, and hence the second term in the parentheses converges to 0. As for the third term, we see

$$e_{m-1}^{(n+1)} = \frac{e_{m-1}^{(n)}}{q_{m-1}^{(n+1)}} \cdot q_m^{(n)} \geq \frac{\sqrt{q_{m-1}^{(n)}} - 2\sqrt{e_{m-1}^{(n)}}}{2q_{m-1}^{(n+1)}} \left(e_{m-1}^{(n)} \right)^{3/2}$$

from (5.14) and (5.6) (with $n+1$ replaced by n). Thus, from (4.2) and (5.5) we see that

$$\lim_{n\to\infty} \frac{e_{m-1}^{(n)}}{\sqrt{e_{m-1}^{(n+1)}}} \leq \lim_{n\to\infty} \left(\frac{\sqrt{q_{m-1}^{(n)}} - 2\sqrt{e_{m-1}^{(n)}}}{2q_{m-1}^{(n+1)}} \right)^{-1/2} \left(e_{m-1}^{(n)} \right)^{1/4} = 0,$$

and hence the value in the parentheses on the right-hand side of (5.15) converges to 1. By using this, together with (5.5), we obtain the claim (5.8):

$$\lim_{n\to\infty} \frac{e_{m-1}^{(n+1)}}{(e_{m-1}^{(n)})^{3/2}} = \lim_{n\to\infty} \frac{\sqrt{q_{m-1}^{(n+1)}}}{q_{m-1}^{(n+2)}} = \frac{1}{\sqrt{{\sigma_{m-1}}^2 - {\sigma_m}^2}}. \tag{5.16}$$

Next, by the second equation in (5.14), and by (5.5) and (5.16), we obtain the claim (5.11):

$$\lim_{n\to\infty} \frac{q_m^{(n)}}{\sqrt{e_{m-1}^{(n)}}} = \lim_{n\to\infty} q_{m-1}^{(n+1)} \frac{e_{m-1}^{(n+1)}}{(e_{m-1}^{(n)})^{3/2}} = \sqrt{{\sigma_{m-1}}^2 - {\sigma_m}^2},$$

which then immediately yields the claim (5.9):

$$\begin{aligned}\lim_{n\to\infty} \frac{q_m^{(n+1)}}{(q_m^{(n)})^{3/2}} &= \lim_{n\to\infty} \frac{q_m^{(n+1)}}{\sqrt{e_{m-1}^{(n+1)}}} \left(\frac{q_m^{(n)}}{\sqrt{e_{m-1}^{(n)}}}\right)^{-3/2} \left(\frac{e_{m-1}^{(n+1)}}{(e_{m-1}^{(n)})^{3/2}}\right)^{1/2} \\ &= ({\sigma_{m-1}}^2 - {\sigma_m}^2)^{1/2-3/4-1/4} \\ &= ({\sigma_{m-1}}^2 - {\sigma_m}^2)^{-1/2}.\end{aligned}$$

Finally, we prove the claims concerning $r_m^{(n)}$. By (3.2), (4.16), and (4.19), we see

$$\lim_{n\to\infty} \frac{r_m^{(n)}}{e_{m-1}^{(n+1)}} = \lim_{n\to\infty} \frac{\sum_{l=n}^{\infty} e_{m-1}^{(l+1)}}{e_{m-1}^{(n+1)}} = \frac{1}{1-\rho_{m-1}} = 1, \tag{5.17}$$

where in the last equality we used the fact that $\rho_{m-1} = 0$, which follows from (4.8) and (5.4). Then, by (4.10) we immediately obtain the claim (5.12):

$$\lim_{n\to\infty} \frac{r_m^{(n)}}{e_{m-1}^{(n)}} = \lim_{n\to\infty} \left(\frac{e_{m-1}^{(n+1)}}{e_{m-1}^{(n)}} \cdot \frac{r_m^{(n)}}{e_{m-1}^{(n+1)}}\right) = \rho_{m-1} = 0.$$

The claim (5.10) also follows from (5.17) as

$$\lim_{n\to\infty} \frac{r_m^{(n+1)}}{(r_m^{(n)})^{3/2}} = \lim_{n\to\infty} \frac{e_{m-1}^{(n+2)}}{(e_{m-1}^{(n+1)})^{3/2}} = \frac{1}{\sqrt{{\sigma_{m-1}}^2 - {\sigma_m}^2}}. \qquad \square$$

6. A numerical experiment. In this section, a simple numerical experiment is presented to illustrate the theory. We consider an $m \times m$ symmetric tridiagonal matrix

$$T = \begin{pmatrix} a & b & & 0 \\ b & a & \ddots & \\ & \ddots & \ddots & b \\ 0 & & b & a \end{pmatrix}, \tag{6.1}$$

the eigenvalues of which are

$$a + 2b\cos\left(\frac{\pi k}{m+1}\right) \quad (k = 1, \ldots, m).$$

The parameters are taken as $m = 10$, $a = 1.0$, and $b = 0.2$, which makes T positive definite. The dqds algorithm is applied to the bidiagonal matrix B obtained from the Cholesky decomposition of T.

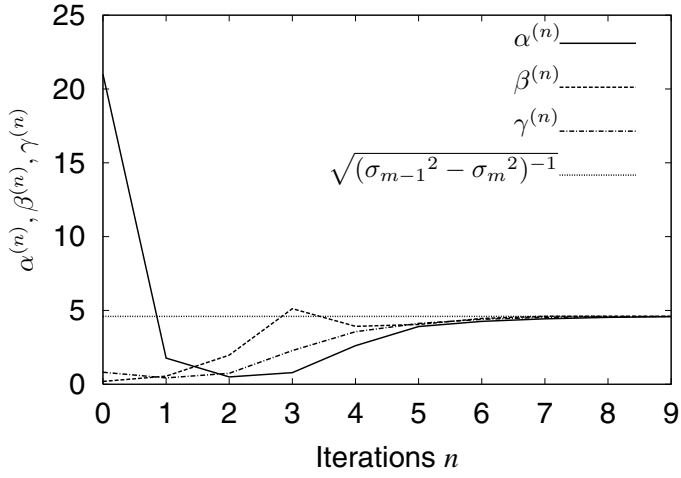


FIG. 2. *Convergence of* $\alpha^{(n)}$, $\beta^{(n)}$, *and* $\gamma^{(n)}$.

TABLE 1
Critical index k^* *for the Johnson bound* (5.1) *in the dqds algorithm.*

n	0	1	2	3	4	5	6	7	8	9
k^*	9	9	10	10	10	10	10	10	10	10

In view of Theorem 5.6, we define

$$\alpha^{(n)} = \frac{e_{m-1}^{(n+1)}}{(e_{m-1}^{(n)})^{3/2}}, \qquad \beta^{(n)} = \frac{q_m^{(n+1)}}{(q_m^{(n)})^{3/2}}, \qquad \gamma^{(n)} = \frac{r_m^{(n+1)}}{(r_m^{(n)})^{3/2}},$$

which should converge to the constant $1/\sqrt{({\sigma_{m-1}}^2 - {\sigma_m}^2)}$ according to the theory. The result is shown in Figure 2. The solid line (——) shows $\alpha^{(n)}$, the dashed line (- - - -) shows $\beta^{(n)}$, and the dashed-dotted line (- · - · -) shows $\gamma^{(n)}$. The dotted line (· · · · · ·) shows $1/\sqrt{({\sigma_{m-1}}^2 - {\sigma_m}^2)} = 4.60$ in this problem setting. The solid line, the dashed line, and the dashed-dotted line approach the dotted line in Figure 2.

In Figure 3, $e_{m-1}^{(n)}$, $q_m^{(n)}$, and $r_m^{(n)}$ are plotted in the single logarithmic graph. The solid line shows $e_{m-1}^{(n)}$, the dashed line shows $q_m^{(n)}$, and the dashed-dotted line shows $r_m^{(n)}$. The variables $e_{m-1}^{(n)}$, $q_m^{(n)}$, and $r_m^{(n)}$ converge to zero. By Figures 2 and 3 we can say that the rate of convergence is 1.5. Table 1 presents the index $k = k^*$, which attains the minimum on the right-hand side of (5.1). If $\lambda^{(n)} < 0$, then k^* is defined to be 0. The result shows that $k^* = m$ for $n \geq 2$, which is consistent with Lemma 5.5.

Acknowledgments. The authors are thankful to Yusaku Yamamoto for a number of helpful comments, and also to the anonymous referees for their constructive suggestions which improved this presentation.

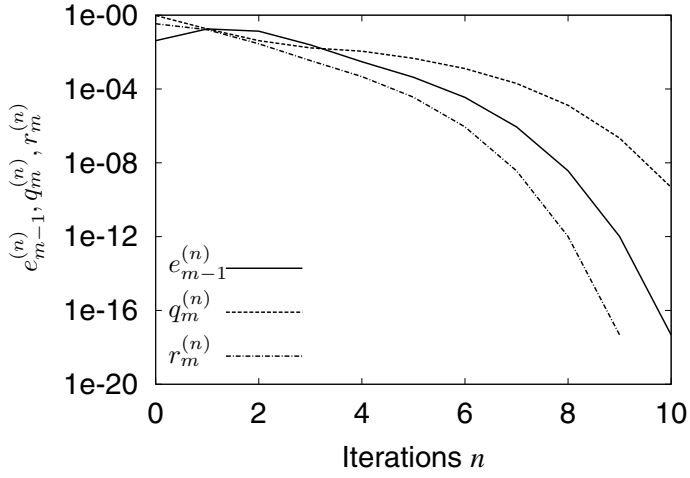


FIG. 3. *Convergence of* $e_{m-1}^{(n)}$, $q_m^{(n)}$, *and* $r_m^{(n)}$.

REFERENCES

[1] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNY, AND D. SORENSEN, *LAPACK Users' Guide*, 3rd ed., SIAM, 1999.

[2] J. DEMMEL AND W. KAHAN, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 873–912.

[3] J. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[4] I. S. DHILLON, *A New* $O(n^2)$ *Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*, Ph.D. thesis, Computer Science Division, University of California, Berkeley, California, 1997.

[5] I. S. DHILLON AND B. N. PARLETT, *Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices*, Linear Algebra Appl., 387 (2004), pp. 1–28.

[6] I. S. DHILLON AND B. N. PARLETT, *Orthogonal eigenvectors and relative gaps*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 858–899.

[7] K. V. FERNANDO AND B. N. PARLETT, *Accurate singular values and differential qd algorithms*, Numer. Math., 67 (1994), pp. 191–229.

[8] P. HENRICI, *Applied and Computational Complex Analysis*, Vol. 1, Power Series—Integration—Conformal Mapping—Location of Zeros, Wiley, New York, 1974.

[9] C. R. JOHNSON, *A Gersgorin-type lower bound for the smallest singular value*, Linear Algebra Appl., 112 (1989), pp. 1–7.

[10] LAPACK, available online at http://www.netlib.org/lapack/.

[11] B. N. PARLETT, *The new qd algorithms*, Acta Numer., 1995, pp. 459–491.

[12] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980; SIAM, Philadelphia, 1998.

[13] B. N. PARLETT AND O. MARQUES, *An implementation of the dqds algorithm (positive case)*, Linear Algebra Appl., 309 (2000), pp. 217–259.

[14] H. RUTISHAUSER, *Solution of eigenvalue problems with the LR-transformation*, Nat. Bur. Standards Appl. Math. Ser., 49 (1958), pp. 47–81.

[15] H. RUTISHAUSER, *Über eine kubisch konvergente Variante der LR-Transformation*, Z. Angew. Math. Mech., 40 (1960), pp. 49–54.

[16] H. RUTISHAUSER, *Lectures on Numerical Mathematics*, Birkhäuser, Boston, 1990.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 538–547

LOW RANK PERTURBATION OF WEIERSTRASS STRUCTURE*

FERNANDO DE TERÁN†, FROILÁN M. DOPICO†, AND JULIO MORO†

Abstract. Let $A_0 + \lambda A_1$ be a regular matrix pencil, and let λ_0 be one of its finite eigenvalues having g elementary Jordan blocks in the Weierstrass canonical form. We show that for most matrices B_0 and B_1 with $\operatorname{rank}(B_0+\lambda_0 B_1) < g$ there are $g-\operatorname{rank}(B_0+\lambda_0 B_1)$ Jordan blocks corresponding to the eigenvalue λ_0 in the Weierstrass form of the perturbed pencil $A_0+B_0+\lambda(A_1+B_1)$. If $\operatorname{rank}(B_0+\lambda_0 B_1)+\operatorname{rank}(B_1)$ does not exceed the number of λ_0-Jordan blocks in $A_0+\lambda A_1$ of dimension greater than one, then the λ_0-Jordan blocks of the perturbed pencil are the $g-\operatorname{rank}(B_0+\lambda_0 B_1)-\operatorname{rank}(B_1)$ smallest λ_0-Jordan blocks of $A_0+\lambda A_1$, together with $\operatorname{rank}(B_1)$ blocks of dimension one. Otherwise, all $g-\operatorname{rank}(B_0+\lambda_0 B_1)$ λ_0-Jordan blocks of the perturbed pencil are of dimension one. This happens for any pair of matrices B_0 and B_1 except those in a proper algebraic submanifold in the set of matrix pairs. If $A_0 + \lambda A_1$ has an infinite eigenvalue, then the corresponding result follows from considering the zero eigenvalue of the dual pencils $A_1 + \lambda A_0$ and $A_1 + B_1 + \lambda(A_0 + B_0)$.

Key words. regular matrix pencils, Weierstrass canonical form, low rank perturbations, matrix spectral perturbation theory

AMS subject classifications. 15A22, 15A18, 15A21

DOI. 10.1137/050633020

1. Introduction.

The change of the Jordan structure of a matrix A under perturbations B of low rank has been recently studied by several authors [5, 7, 8, 9, 10]. It is known that if λ_0 is one of the eigenvalues of A having g elementary Jordan blocks in the Jordan canonical form of A, then [illegible] B satisfying $\operatorname{rank}(B) < g$, the Jordan blocks of $A + B$ with eigenvalue λ_0 are just the $g - \operatorname{rank}(B)$ smallest Jordan blocks of A with eigenvalue λ_0. As far as we know, this generic behavior was first proved in [5] and again in [7] and [8, 9, 10]. The proof in [7] uses only elementary linear algebra results, and allows us to explicitly characterize the set of perturbation matrices B for which this generic behavior does not happen. This is done through a scalar determinantal equation involving B and some of the λ_0-eigenvectors of A. Thus, this behavior can be properly termed as [illegible], since it happens for any perturbation matrix B except those belonging to a proper algebraic submanifold in the set of $n \times n$ matrices of given rank. It is interesting to note that the result in [5] remains valid for infinite dimensional compact linear operators in Banach spaces.

The purpose of this paper is to study which is the generic change of the Weierstrass canonical form [4] of a regular $n \times n$ pencil of matrices $A_0 + \lambda A_1$ under a low rank perturbation $B_0 + \lambda B_1$. We will see that this change is rather different from the change described above for matrices. The regular matrix pencil $A_0+\lambda A_1$ may have an [illegible] eigenvalue, whose Jordan blocks in the Weiertrass canonical form are precisely the Jordan blocks associated with the [illegible] eigenvalue in the Weierstrass form of the [illegible] pencil $A_1 + \lambda A_0$. Therefore, we may focus on finite eigenvalues of $A_0 + \lambda A_1$. The perturbation results for the infinite eigenvalue follow from results for the zero eigenvalue of the dual pencil.

*Received by the editors June 3, 2005; accepted for publication (in revised form) by B. T. Kågström January 3, 2008; published electronically May 16, 2008. This research was partially supported by the Ministerio de Educación y Ciencia of Spain through grant BFM 2003-00223.
http://www.siam.org/journals/simax/30-2/63302.html

†Departamento de Matemáticas, Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911-Leganés, Madrid, Spain (fteran@math.uc3m.es, dopico@math.uc3m.es, jmoro@math.uc3m.es).

Let λ_0 be a finite eigenvalue with geometric multiplicity g of the regular $n \times n$ matrix pencil $A_0 + \lambda A_1$. Recall that a pencil is regular if the polynomial $\det(A_0 + \lambda A_1)$ in λ is not identically zero, and that the geometric multiplicity of λ_0 is $g = \dim \ker(A_0 + \lambda_0 A_1)$, where ker denotes the null space. The elementary inequalities $\operatorname{rank}(C + D) \leq \operatorname{rank}(C) + \operatorname{rank}(D)$ and $\operatorname{rank}(C) \leq \operatorname{rank}(C + D) + \operatorname{rank}(D)$, valid for any pair of matrices C and D, lead to

$$\operatorname{rank}(A_0 + \lambda_0 A_1 + B_0 + \lambda_0 B_1) \leq \operatorname{rank}(A_0 + \lambda_0 A_1) + \operatorname{rank}(B_0 + \lambda_0 B_1),$$
$$\operatorname{rank}(A_0 + \lambda_0 A_1) \leq \operatorname{rank}(A_0 + \lambda_0 A_1 + B_0 + \lambda_0 B_1) + \operatorname{rank}(B_0 + \lambda_0 B_1).$$

Combining both inequalities, one gets

$$g - \operatorname{rank}(B_0 + \lambda_0 B_1) \leq \dim \ker(A_0 + \lambda_0 A_1 + B_0 + \lambda_0 B_1) \leq g + \operatorname{rank}(B_0 + \lambda_0 B_1). \tag{1.1}$$

Therefore, whenever

$$\operatorname{rank}(B_0 + \lambda_0 B_1) < g, \tag{1.2}$$

the eigenvalue λ_0 of $A_0 + \lambda A_1$ stays as an eigenvalue of the perturbed pencil

$$A_0 + B_0 + \lambda(A_1 + B_1). \tag{1.3}$$

As a consequence, by "low" rank perturbation we will mean in what follows that B_0 and B_1 satisfy (1.2), a condition which depends on the particular eigenvalue λ_0 we are considering. It is well known that for a regular pencil $L_0 + \lambda L_1$ the number of Jordan blocks associated with λ_0 in its Weierstrass canonical form is equal to $\dim \ker(L_0 + \lambda_0 L_1)$. Therefore, assuming that (1.3) is still regular, (1.1) implies that the perturbation $B_0 + \lambda B_1$ can destroy at most $\operatorname{rank}(B_0 + \lambda_0 B_1)$ Jordan blocks of $A_0 + \lambda A_1$, and can create at most $\operatorname{rank}(B_0 + \lambda_0 B_1)$ new Jordan blocks associated with the finite eigenvalue λ_0 of $A_0 + \lambda A_1$. This allows many different choices for the number and dimensions of the Jordan blocks appearing in the Weierstrass form of $A_0 + B_0 + \lambda(A_1 + B_1)$. The goal of this work is to find out which is the generic behavior in this respect.[1]

The result we present depends on two quantities [illegible] λ_0, namely

$$\rho_0 = \operatorname{rank}(B_0 + \lambda_0 B_1) \quad \text{and} \quad \rho_1 = \operatorname{rank}(B_1).$$

Assuming that condition (1.2) holds, we will prove that [illegible] B_0, B_1 there are precisely $g - \rho_0$ Jordan blocks associated with λ_0 in the Weierstrass canonical form of the perturbed pencil (1.3). Moreover, if we denote by d_0 the number of Jordan blocks in $A_0 + \lambda A_1$ with eigenvalue λ_0 of dimension greater than one, we will prove that whenever $\rho_0 + \rho_1 \leq d_0$, the largest ρ_0 Jordan blocks of $A_0 + \lambda A_1$ associated with λ_0 disappear, and the second-largest ρ_1 blocks of λ_0 turn into 1×1 blocks, while the rest of the Jordan blocks of λ_0 in $A_0 + \lambda A_1$ remain as Jordan blocks in the perturbed pencil (1.3). If $\rho_0 + \rho_1 > d_0$, then there will be only 1×1 blocks corresponding to λ_0 in the Weierstrass form of (1.3). This generic behavior coincides with the one previously described for low rank perturbations of the Jordan canonical form of matrices in the case $B_1 = 0$, while it is rather different when $B_1 \neq 0$. [illegible]

[1] The assumption that $A_0 + B_0 + \lambda(A_1 + B_1)$ is a regular pencil holds except for very particular choices of B_0 and B_1.

Inequality (1.1) makes clear that $B_0 + \lambda_0 B_1$ is bound to play a relevant role in the perturbation of the Weierstrass structure, since it determines the geometric multiplicity of λ_0 in (1.3). To understand why B_1 plays a separate role on its own, recall that a Jordan chain of $A_0 + \lambda A_1$ of length s associated with λ_0 satisfies the equations $(A_0 + \lambda_0 A_1)v_1 = 0$ and $(A_0 + \lambda_0 A_1)v_k = A_1 v_{k-1}$ for $2 \leq k \leq s$. Therefore, it is expected that perturbing A_1 affects to the length of the Jordan chains. In plain words, the generic behavior described above corresponds to a cooperation between $B_0 + \lambda_0 B_1$ and B_1 to destroy some of the blocks, and to decrease the dimension of as many of the largest Jordan blocks as possible, while still fulfilling the constraint (1.1) on the geometric multiplicity.

The results obtained in the present paper, as those in [7], are valid for perturbations of [illegible] satisfying the low rank condition (1.2), i.e., they are not first-order perturbation results. Notice also that we are not paying attention to the perturbation of the eigenvalues corresponding to the destroyed Jordan blocks. First order perturbation results for this problem are enumerated in [6] for general matrix polynomials, and, more recently, in [2]. In [13] first order multiparametric perturbations have been considered for multiple semisimple eigenvalues. Several perturbation bounds, valid for perturbations of finite size, appear in [11], but they do not apply to multiple defective eigenvalues, except in the case of some Gerschgorin-like inclusion regions.

Now, we summarize the [illegible] of a regular pencil [4], and introduce some notation to be used throughout the paper. For any regular $n \times n$ complex matrix pencil $A_0 + \lambda A_1$ having λ_0 as one of its eigenvalues, there exist nonsingular $n \times n$ matrices P and Q, independent of λ, such that

$$Q(A_0 + \lambda A_1)P = \operatorname{diag}(J_{n_1}(-\lambda_0), \ldots, J_{n_g}(-\lambda_0), \widetilde{J}, I_\infty) + \lambda \operatorname{diag}(I_1, I_2, N), \tag{1.4}$$

where $\operatorname{diag}(C, E)$ denotes a block diagonal matrix with square diagonal blocks C and E; $J_{n_i}(-\lambda_0)$ stands for a Jordan block of dimension n_i with $-\lambda_0$ on the main diagonal; $\widetilde{J}$ is a matrix in Jordan canonical form corresponding to the other finite eigenvalues of the pencil; and N is a matrix in Jordan canonical form whose eigenvalues are all equal to zero. N contains the spectral structure of the infinite eigenvalue of the pencil. Finally, I_1, I_2 and I_∞ are identity matrices of matching dimensions to those of $\operatorname{diag}(J_{n_1}(-\lambda_0), \ldots, J_{n_g}(-\lambda_0))$, $\widetilde{J}$ and N, respectively. The right-hand side of (1.4) is the Weierstrass canonical form of the pencil $A_0 + \lambda A_1$, and it is unique up to permutation of the diagonal Jordan blocks. The Weierstrass canonical form displays all of the spectral information of the regular pencil $A_0 + \lambda A_1$. From (1.4), one can easily see that the [illegible] of λ_0 is g and its [illegible] is

$$a_{A_0+\lambda A_1}(\lambda_0) = n_1 + \cdots + n_g. \tag{1.5}$$

Without loss of generality, we assume the dimensions n_i to be ordered decreasingly, i.e.,

$$n_1 \geq n_2 \geq \cdots \geq n_g. \tag{1.6}$$

The paper is organized as follows: Section 2 contains one of the main results (Theorem 2.2) concerning the change of the Weierstrass structure. It gives a lower bound on the algebraic multiplicity associated with each eigenvalue in the perturbed pencil, and suggests that the generic behavior for the Jordan blocks of the perturbed pencil is the one happening when this lower bound is attained. In section 3, we prove that this behavior is indeed generic by showing that it holds for all perturbations except those in a proper algebraic submanifold in the set of matrix pencils. Finally, Theorem 3.3 summarizes the results obtained throughout this paper.

2. Lower bounds on the algebraic multiplicities and the dimensions of Jordan blocks in the perturbed pencil. Throughout this section we follow a notation consistent with (1.5), and denote by

$$a_{R(\lambda)}(\lambda_0) \tag{2.1}$$

the algebraic multiplicity of the eigenvalue λ_0 in the regular matrix pencil $R(\lambda)$. Our aim is to determine the generic Weierstrass structure of λ_0 in a perturbed matrix pencil $A_0 + B_0 + \lambda(A_1 + B_1)$, starting from the structure of this eigenvalue in the unperturbed pencil $A_0 + \lambda A_1$. For this, we need to know $a_{A_0+B_0+\lambda(A_1+B_1)}(\lambda_0)$, as well as how this algebraic multiplicity is distributed among the Jordan blocks of λ_0. At least two approaches are possible to solve this problem. First, one can start with Jordan chains of $A_0 + \lambda A_1$ associated with λ_0, and then explicitly build new Jordan chains for λ_0 in $A_0 + B_0 + \lambda(A_1 + B_1)$, exhausting the algebraic multiplicity $a_{A_0+B_0+\lambda(A_1+B_1)}(\lambda_0)$. This approach was used in [7, 8, 9] for the standard eigenvalue problem $A - \lambda I$. It has the advantage of providing the new Jordan chains, and the drawback of being rather intricate in the case of pencils. In this paper we use a simpler approach: First, we determine lower bounds on the number and the dimensions of the Jordan blocks associated with λ_0 in the perturbed pencil. This method, based on a result by Thompson [12], involves the use of the invariant factors of the pencils. Then, we prove that the generic behavior corresponds to the case when these lower bounds are attained.

We begin by recalling that the rank of an arbitrary matrix pencil, regular or singular, $T(\lambda) = T_0 + \lambda T_1$ is r if all of the minors of $T(\lambda)$ of dimension greater than r are identically equal to zero, but $T(\lambda)$ has minors of dimension r which are polynomials in λ not identically equal to zero. As a consequence, the rank of a regular $n \times n$ matrix pencil is equal to n.

The next auxiliary lemma is a consequence of [12, Theorem 1]. It establishes lower bounds on the number and dimensions of the Jordan blocks in the Weierstrass form of a regular matrix pencil $(R+T)(\lambda)$, where $R(\lambda)$ is a regular matrix pencil and $T(\lambda)$ is any pencil of rank r.

LEMMA 2.1. $R(\lambda) = R_0 + \lambda R_1$ $T(\lambda) = T_0 + \lambda T_1$ r λ_0 $R(\lambda)$ g $d_1 \geq \cdots \geq d_g$ $R(\lambda)$ $(R + T)(\lambda)$ $r \leq g$ $(R+T)(\lambda)$ $g - r$ λ_0 $\beta_{r+1} \geq \cdots \geq \beta_g$ $\beta_i \geq d_i$ $r + 1 \leq i \leq g$

. First, let us assume that the rank of $T(\lambda)$ is exactly r. We begin by proving that any pencil $T(\lambda)$ of rank r is the sum of r singular pencils of rank 1. This can be seen by using the Kronecker canonical form of singular pencils [4, Chapter XII]. Let $K_0 + \lambda K_1$ be the Kronecker canonical form of $T(\lambda) = T_0 + \lambda T_1$, and write $K_0 + \lambda K_1$ as the sum of the following matrices:

1. For any singular block L_k of dimension $k \times (k + 1)$ appearing in $K_0 + \lambda K_1$ [4, p. 39], we have that $L_k = L_k^{(1)} + \cdots + L_k^{(k)}$, where the jth row of $L_k^{(j)}$ is equal to the jth row of L_k, and the rest of the rows of $L_k^{(j)}$ are zero. Therefore L_k is the sum of k singular pencils with rank 1.

2. An analogous expression holds for any singular block L_p^T of dimension $(p + 1) \times p$ appearing in $K_0 + \lambda K_1$.

3. Finally, for the $m \times m$ regular part $F(\lambda) = F_0 + \lambda F_1$ of $K_0 + \lambda K_1$, we have again that $F(\lambda) = F^{(1)} + \cdots + F^{(m)}$, where $F^{(j)}$ has the jth row equal to the jth row of $F(\lambda)$ and the rest of the rows of $F^{(j)}$ are zero. Therefore $F(\lambda)$ is the sum of m singular pencils with rank 1.

The pencil $K_0 + \lambda K_1$ can be expressed as the sum of r singular pencils of rank 1 just by combining the previous expansions of its singular blocks and of its regular part. The same holds for $T(\lambda)$ because it is strictly equivalent to $K_0 + \lambda K_1$. Let this decomposition be

$$T(\lambda) = T_1(\lambda) + \cdots + T_r(\lambda),$$

where $\operatorname{rank} T_i(\lambda) = 1$ for $1 \leq i \leq r$.

For any $n \times n$ regular pencil $P(\lambda) = P_0 + \lambda P_1$, we denote by

$$h_n(P)|h_{n-1}(P)|\cdots|h_1(P)$$

its invariant polynomials [3, Chapter VI], also called invariant factors. As usual, $h_n(P)|h_{n-1}(P)$ means that $h_n(P)$ divides $h_{n-1}(P)$. Notice also that $h_1(P) \neq 0$ because the pencil is regular.

Let

$$(\lambda - \lambda_0)^{d_g}|(\lambda - \lambda_0)^{d_{g-1}}|\cdots|(\lambda - \lambda_0)^{d_1}$$

be the elementary divisors [3, Chapter VI] of $R(\lambda)$ associated with λ_0. Each elementary divisor $(\lambda - \lambda_0)^{d_i}$ corresponds to a Jordan block of λ_0 of dimension d_i in the Weierstrass form of $R(\lambda)$. It is well known that

$$\begin{array}{c} (\lambda - \lambda_0)^{d_1}|h_1(R) \\ (\lambda - \lambda_0)^{d_2}|h_2(R) \\ \vdots \quad \vdots \quad \vdots \\ (\lambda - \lambda_0)^{d_g}|h_g(R). \end{array}$$

Now, consider the sequence of pencils $R(\lambda), R(\lambda)+T_1(\lambda), R(\lambda)+T_1(\lambda)+T_2(\lambda), \ldots, R(\lambda) + T(\lambda)$, and note that each of them is a rank 1 perturbation of the preceding one. Applying [12, Theorem 1][2] to this sequence leads to

$$\begin{array}{l} (\lambda - \lambda_0)^{d_{r+1}}|h_{r+1}(R)|h_r(R+T_1)|\ldots|h_1(R+T), \\ (\lambda - \lambda_0)^{d_{r+2}}|h_{r+2}(R)|h_{r+1}(R+T_1)|\ldots|h_2(R+T), \\ \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \quad \vdots \\ (\lambda - \lambda_0)^{d_g}|h_g(R)|h_{g-1}(R+T_1)|\ldots|h_{g-r}(R+T), \end{array}$$

where $h_1(R+T) \neq 0$ because the pencil $(R+T)(\lambda)$ is regular. These divisibility chains mean that the pencil $(R+T)(\lambda)$ has $g-r$ elementary divisors associated with λ_0:

$$(\lambda - \lambda_0)^{\beta_g}|(\lambda - \lambda_0)^{\beta_{g-1}}|\cdots|(\lambda - \lambda_0)^{\beta_{r+1}},$$

[2]Theorem 1 in [12] states that if $P(\lambda)$ and $Q(\lambda)$ are $n \times n$ matrix polynomials with invariant polynomials $h_n(P)|h_{n-1}(P)|\cdots|h_1(P)$ and $h_n(Q)|h_{n-1}(Q)|\cdots|h_1(Q)$, respectively, and if the rank of $P(\lambda) - Q(\lambda)$ is equal to one, then $h_n(P)|h_{n-1}(Q)|h_{n-2}(P)|h_{n-3}(Q)|\cdots$ and $h_n(Q)|h_{n-1}(P)|h_{n-2}(Q)|h_{n-3}(P)|\cdots$.

with $d_i \leq \beta_i$ for $r+1 \leq i \leq g$. Each of these elementary divisors corresponds to a $\beta_i \times \beta_i$ Jordan block associated with λ_0 in the Weierstrass form of $(R+T)(\lambda)$.

If the rank of $T(\lambda)$ is $r_1 < r$, then the result we have just proved can be applied to show that the Weierstrass form of $(R+T)(\lambda)$ has at least $g - r_1 > g - r$ Jordan blocks associated with λ_0 of dimensions $\beta_i \geq d_i$, $i = r_1 + 1, \ldots, g$, and the result follows. □

The previous lemma allows us to obtain the main result in the first part of the present paper.

THEOREM 2.2. [illegible] λ_0 [illegible] $A_0 + \lambda A_1$ [illegible] $n_1 \geq \cdots \geq n_g$ [illegible] λ_0 [illegible] $B_0 + \lambda B_1$ [illegible] $A_0 + B_0 + \lambda(A_1 + B_1)$ [illegible] $g \geq \operatorname{rank}(B_0 + \lambda_0 B_1)$ [illegible] $\rho = \operatorname{rank}(B_0 + \lambda_0 B_1) + \operatorname{rank} B_1$ [illegible] $n_m = 1$ [illegible] $m = g+1, \ldots, \rho$ [illegible] λ_0 [illegible]

$$a_{A_0+B_0+\lambda(A_1+B_1)}(\lambda_0) \geq a_{A_0+\lambda A_1}(\lambda_0) + \operatorname{rank} B_1 - n_1 - \cdots - n_\rho, \tag{2.2}$$

[illegible] (2.1) [illegible] λ_0 [illegible] $A_0 + B_0 + \lambda(A_1 + B_1)$ [illegible] ρ [illegible] $n_1, \ldots, n_g, \underbrace{1, \ldots, 1}_{\operatorname{rank} B_1}$ [illegible]. Notice that

$$\operatorname{rank}(B_0 + \lambda B_1) = \operatorname{rank}(B_0 + \lambda_0 B_1 + (\lambda - \lambda_0) B_1) \leq \operatorname{rank}(B_0 + \lambda_0 B_1) + \operatorname{rank}(B_1) = \rho.$$

So, in the case $\rho < g$, Lemma 2.1 guarantees the existence of $g - \rho$ Jordan blocks associated with λ_0 of dimensions $\beta_{\rho+1} \geq n_{\rho+1}, \ldots, \beta_g \geq n_g$ in the Weierstrass canonical form of the perturbed pencil $A_0 + B_0 + \lambda(A_1 + B_1)$. Moreover, the left side in the inequality (1.1) implies that there are at least $\rho_1 = \operatorname{rank} B_1$ additional Jordan blocks of sizes $\alpha_1 \geq 1, \ldots, \alpha_{\rho_1} \geq 1$ associated with λ_0. Thus,

$$a_{A_0+B_0+\lambda(A_1+B_1)})(\lambda_0) \geq \beta_{\rho+1} + \cdots + \beta_g + \alpha_1 + \cdots + \alpha_{\rho_1} \geq n_{\rho+1} + \cdots + n_g + \rho_1.$$

Obviously, this inequality is equivalent to (2.2). If $g \leq \rho$, then inequality (2.2) becomes

$$a_{A_0+B_0+\lambda(A_1+B_1)}(\lambda_0) \geq g - \operatorname{rank}(B_0 + \lambda_0 B_1).$$

This fact is trivial because of inequality (1.1) and the evident spectral inequality

$$a_{A_0+B_0+\lambda(A_1+B_1)}(\lambda_0) \geq \dim \ker (A_0 + \lambda_0 A_1 + B_0 + \lambda_0 B_1).$$

Finally, notice that the previous inequalities become equalities if and only if the number and dimensions of the Jordan blocks associated with λ_0 in the Weierstrass form of $A_0 + B_0 + \lambda(A_1 + B_1)$ are those appearing in the statement of Theorem 2.2. □

[illegible] 1. Notice that the unnatural definition $n_m = 1$ for $m = g+1, \ldots, \rho$ allows us to express inequality (2.2) in a unified way for both cases $\rho < g$ and $\rho \geq g$. The reader is invited to check that the number and dimensions of the Jordan blocks of the perturbed pencil associated with λ_0 in the case of equality in (2.2) are

precisely those appearing in the generic behavior described in the abstract and the introduction.

Theorem 2.2 gives us all the sizes of the Jordan blocks associated with λ_0 in the perturbed pencil $A_0+B_0+\lambda(A_1+B_1)$ when the inequality in (2.2) is an equality. As we will see in the following section, this is the case for most perturbations $B_0+\lambda B_1$.

3. The generic behavior. The quantity

$$\widetilde{a} = a_{A_0+\lambda A_1}(\lambda_0) + \text{rank } B_1 - n_1 - \cdots - n_\rho$$

in (2.2), where $\rho = \text{rank}\,(B_0+\lambda_0 B_1) + \text{rank } B_1$ as in the statement of Theorem 2.2, is a lower bound on the algebraic multiplicity of λ_0 as an eigenvalue of the perturbed pencil $A_0+B_0+\lambda(A_1+B_1)$. This means that for each perturbation $B_0+\lambda B_1$ of $A_0+\lambda A_1$ such that $g \geq \text{rank}(B_0+\lambda_0 B_1)$,

$$\det\,(A_0+B_0+\lambda(A_1+B_1)) = (\lambda-\lambda_0)^{\widetilde{a}} q(\lambda-\lambda_0) \tag{3.1}$$

for some polynomial $q(\lambda-\lambda_0)$. Therefore, if the perturbed pencil is regular the algebraic multiplicity of λ_0 in the perturbed pencil is exactly $\widetilde{a}$ if and only if the coefficient $q(0)$ of $(\lambda-\lambda_0)^{\widetilde{a}}$ in (3.1) is not equal to zero. Clearly, once A_0 and A_1 are fixed, this coefficient is a multivariate polynomial in the entries of B_0 and B_1. Therefore, if this coefficient is not identically zero for all B_0 and B_1 such that $\text{rank}(B_0+\lambda_0 B_1) = \rho_0 \leq g$ and $\text{rank}(B_1) = \rho_1$, for fixed integers ρ_0 and ρ_1, the equation $q(0)=0$ defines an algebraic submanifold in the set of pairs (B_0, B_1) with $\text{rank}(B_0+\lambda_0 B_1) = \rho_0 \leq g$ and $\text{rank}(B_1) = \rho_1$ that characterizes the set of perturbation pencils for which the generic behavior described in the introduction does not happen. The only goal of this section is to show that this algebraic submanifold is proper or, in other words, that the coefficient $q(0)$ is not zero for all perturbations $B_0+\lambda B_1$ such that $\text{rank}(B_0+\lambda_0 B_1) = \rho_0 \leq g$ and $\text{rank}(B_1) = \rho_1$. This is done in Lemma 3.2. This will allow us to say that the change in the dimensions of the Jordan blocks described in Theorem 2.2, when the equality in (2.2) holds, is [illegible]. The reader is referred to [1] for a detailed description of the algebraic submanifold $q(0)=0$ in terms of a determinantal equation involving the entries of B_0 and B_1.

The simple Lemma 3.1 studies some specific perturbations of the blocks appearing in the Weierstrass canonical form (1.4). It will be used in the proof of Lemma 3.2.

LEMMA 3.1. [illegible] $J_k(\alpha)$ [illegible] $k\times k$ [illegible] α [illegible] $E_k(\beta)$ [illegible] $k\times k$ [illegible] β [illegible] $(k,1)$ [illegible] $D_k(\lambda) = (\lambda-\lambda_0)\,\text{diag}(s_1,\ldots,s_k)$ [illegible] $s_i = 0$ [illegible] 1 [illegible] i [illegible] $D_k(\lambda)$ [illegible]

1. λ_0 [illegible] $\lambda I + J_k(-\lambda_0) + E_k(\lambda-\lambda_0) + D_k(\lambda)$ [illegible] 1
2. λ_0 [illegible] $\lambda I + J_k(-\lambda_0) + E_k(1) + D_k(\lambda)$
3. λ_0 [illegible] $\lambda I + J_k(-\lambda_1) + D_k(\lambda)$ [illegible] $\lambda_1 \neq \lambda_0$
4. λ_0 [illegible] $\lambda J_k(0) + I + D_k(\lambda)$

[illegible] Check that

1.

$$\begin{aligned}&\det(\lambda I + J_k(-\lambda_0) + E_k(\lambda-\lambda_0) + D_k(\lambda))\\ &\qquad = (\lambda-\lambda_0)\left[(\lambda-\lambda_0)^{k-1}\left(\prod_{i=1}^{k}(1+s_i)\right) + (-1)^{k+1}\right],\end{aligned}$$

2.

$$\det(\lambda I + J_k(-\lambda_0) + E_k(1) + D_k(\lambda)) = (\lambda - \lambda_0)^k \left(\prod_{i=1}^{k} (1 + s_i) \right) + (-1)^{k+1},$$

3.

$$\det(\lambda I + J_k(-\lambda_1) + D_k(\lambda)) = \prod_{i=1}^{k} \left[(\lambda - \lambda_1) + s_i(\lambda - \lambda_0) \right],$$

4.

$$\det(\lambda J_k(0) + I + D_k(\lambda)) = \prod_{i=1}^{k} \left[1 + s_i(\lambda - \lambda_0) \right]. \qquad \square$$

LEMMA 3.2. [illegible] λ_0 [illegible] $n \times n$ [illegible] $A_0 + \lambda A_1$ [illegible] $n_1 \geq \cdots \geq n_g$ [illegible] λ_0 [illegible] ρ_0 [illegible] ρ_1 [illegible] $\rho_0 \leq g$ [illegible] $\rho_1 \leq n$ [illegible] $B_0 + \lambda B_1$ [illegible]

$$\rho_0 = \operatorname{rank}(B_0 + \lambda_0 B_1), \qquad \rho_1 = \operatorname{rank}(B_1),$$

[illegible] λ_0 [illegible] $A_0 + B_0 + \lambda(A_1 + B_1)$ [illegible] $a_{A_0+\lambda A_1}(\lambda_0) + \rho_1 - n_1 - \cdots - n_\rho$ [illegible] $\rho := \rho_0 + \rho_1$ [illegible] $n_m = 1$ [illegible] $m = g + 1, \ldots, \rho$

[illegible] It suffices to prove the result when $A_0 + \lambda A_1$ is in Weierstrass canonical form because, otherwise, we can consider the strict equivalence (1.4), apply the result to the matrix pencil in Weierstrass canonical form in the right-hand side (with $B_0 + \lambda B_1$ as the perturbation pencil), and take $Q^{-1}(B_0 + \lambda B_1)P^{-1}$.

So, assume that $A_0 + \lambda A_1$ is in Weierstrass canonical form given by the right-hand side of (1.4). We consider separately the following two cases.

(i) Case $\rho < g$. Define the matrices

$$B_0 = \operatorname{diag}(E_{n_1}(1), \ldots, E_{n_{\rho_0}}(1), E_{n_{\rho_0+1}}(-\lambda_0), \ldots, E_{n_{\rho_0+\rho_1}}(-\lambda_0), 0, \ldots, 0)$$

and

$$B_1 = \operatorname{diag}(\overbrace{0, \ldots, 0}^{\rho_0 \ blocks}, E_{n_{\rho_0+1}}(1), \ldots, E_{n_{\rho_0+\rho_1}}(1), 0, \ldots, 0),$$

where zeros denote matrices, and the partition in diagonal blocks is conformal to the one of the Weierstrass form (1.4). It can be checked that the pencil $B_0 + \lambda B_1$ verifies the conditions mentioned in the statement, by using the first two items in Lemma 3.1 with $D_k(\lambda) = 0$.

(ii) Case $\rho \geq g$. Now, we define

$$\widehat{B}_0 = \operatorname{diag}(E_{n_1}(1), \ldots, E_{n_{\rho_0}}(1), E_{n_{\rho_0+1}}(-\lambda_0), \ldots, E_g(-\lambda_0), 0, \ldots, 0)$$

and

$$\widehat{B}_1 = \operatorname{diag}(\overbrace{0, \ldots, 0}^{\rho_0 \ blocks}, E_{n_{\rho_0+1}}(1), \ldots, E_{n_g}(1), 0, \ldots, 0),$$

where the partition is again conformal to the one in the Weierstrass canonical form (1.4). Notice that $\operatorname{rank}(\widehat{B}_1) = g - \rho_0 \leq \rho_1$, and that by appropriate choices of $\{s_1, \ldots, s_n\}$, $s_i = 0$ or 1 for all i, $\operatorname{rank}(\widehat{B}_1 + \operatorname{diag}(s_1, \ldots, s_n))$ may take any value between $g - \rho_0$ and n. Let $\{\tilde{s}_1, \ldots, \tilde{s}_n\}$ be such that $\operatorname{rank}(\widehat{B}_1 + \operatorname{diag}(\tilde{s}_1, \ldots, \tilde{s}_n)) = \rho_1$, and define the pencil $D(\lambda) = D_0 + \lambda D_1 = (\lambda - \lambda_0)\operatorname{diag}(\tilde{s}_1, \ldots, \tilde{s}_n)$. Then the pencil $B_0 + \lambda B_1 \equiv \widehat{B}_0 + \lambda \widehat{B}_1 + D(\lambda)$ verifies the conditions mentioned in the statement, because $\operatorname{rank}(B_1) = \rho_1$,

$$\operatorname{rank}(B_0 + \lambda_0 B_1) = \operatorname{rank}(\widehat{B}_0 + \lambda_0 \widehat{B}_1) = \rho_0,$$

and Lemma 3.1 implies that the algebraic multiplicity of λ_0 in $A_0 + B_0 + \lambda(A_1 + B_1)$ is $g - \rho_0$, which is exactly $a_{A_0+\lambda A_1}(\lambda_0) + \rho_1 - n_1 - \cdots - n_\rho$. ∎

Theorem 2.2 and Lemma 3.2 allow us to give a complete answer to the problem originally posed in the introduction: Given a regular pencil $A_0 + \lambda A_1$ with eigenvalue λ_0, perturbed by a pencil $B_0 + \lambda B_1$, determine the generic Weierstrass structure associated with λ_0 as an eigenvalue of the perturbed pencil (1.3) when the low rank condition (1.2) holds for the perturbation. Notice that if $B_0 + \lambda B_1$ is in the set consisting of perturbation pencils for which $q(0) \neq 0$ (with $q(\lambda)$ as in (3.1)), then the perturbed pencil $A_0 + B_0 + \lambda(A_1 + B_1)$ is regular. With this observation in mind we can state the main theorem of this paper in the following way.

THEOREM 3.3. [illegible] λ_0 [illegible] $n \times n$ [illegible] $A_0 + \lambda A_1$ [illegible] (1.4) [illegible] g [illegible] λ_0 [illegible] $A_0 + \lambda A_1$ [illegible] $B_0 + \lambda B_1$ [illegible] $n \times n$ [illegible]

$$\rho_0 := \operatorname{rank}(B_0 + \lambda_0 B_1), \qquad \rho_1 := \operatorname{rank}(B_1), \qquad \rho := \rho_0 + \rho_1.$$

[illegible] $\rho_0 < g$ [illegible] λ_0 [illegible]

$$A_0 + B_0 + \lambda(A_1 + B_1), \tag{3.2}$$

[illegible] $A_0 + B_0 + \lambda(A_1 + B_1)$ [illegible] λ_0 [illegible] $A_0 + B_0 + \lambda(A_1 + B_1)$ [illegible] ρ [illegible] $n_1, \ldots, n_g, \underbrace{1, \ldots, 1}_{\rho_1}$

[illegible] 2. 1. An analogous result holds for the infinite eigenvalue of $A_0 + \lambda A_1$, by applying the previous theorem to the zero eigenvalue of the dual pencils $A_1 + \lambda A_0$ and $A_1 + B_1 + \lambda(A_0 + B_0)$.

2. Theorem 3.3 describes in a concise way the generic behavior presented in the introduction of this paper.

Acknowledgments. The third author thanks Prof. Sergey Savchenko for bringing to his attention Theorem 1 in [12] and the reference [5]. The authors are also indebted to an anonymous referee for valuable comments which very much improved the original manuscript.

REFERENCES

[1] F. DE TERÁN, *Problemas de perturbación de objetos espectrales discontinuos en haces matriciales*, Ph.D. Dissertation, Universidad Carlos III de Madrid, Madrid, Spain, 2007 (in Spanish, available upon request to the author).

[2] F. DE TERÁN, F. M. DOPICO, AND J. MORO, *First order spectral perturbation theory of square singular matrix pencils*, Linear Algebra Appl., submitted.

[3] F. R. Gantmacher, *The Theory of Matrices, Vol.* I, Chelsea Publishing Co., New York, 1959.
[4] F. R. Gantmacher, *The Theory of Matrices, Vol.* II, Chelsea Publishing Co., New York, 1959.
[5] L. Hörmander and A. Melin, *A remark on perturbations of compact operators*, Math. Scand., 75 (1994), pp. 255–262.
[6] H. Langer and B. Najman, *Remarks on the perturbation of analytic matrix functions*, III, Integral Equations Operator Theory, 15 (1992), pp. 796–806.
[7] J. Moro and F. M. Dopico, *Low rank perturbation of Jordan structure*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 495–506.
[8] S. V. Savchenko, *On the typical change of the spectral properties under a rank-one perturbation*, Mat. Zametki, 74 (2003), pp. 590–602 (in Russian).
[9] S. V. Savchenko, *On the change in the spectral properties of a matrix under perturbations of sufficiently low rank*, Funkts. Anal. Prilozh., 38 (2004), pp. 85–88 (in Russian). Translation in Funct. Anal. Appl., 38 (2004), pp. 69–71.
[10] S. V. Savchenko, *Laurent expansion for the determinant of the matrix of scalar resolvents*, Mat. Sb., 196 (2005), pp. 121–144 (in Russian). Translation in Sb. Math., 196 (2005), pp. 743–764.
[11] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
[12] R. C. Thompson, *Invariant factors under rank one perturbations*, Canad. J. Math., 32 (1980), pp. 240–245.
[13] H. Xie and H. Dai, *On the sensitivity of multiple eigenvalues of nonsymmetric matrix pencils*, Linear Algebra Appl., 374 (2003), pp. 143–158.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 548–559

BOUNDS ON CHANGES IN RITZ VALUES FOR A PERTURBED INVARIANT SUBSPACE OF A HERMITIAN MATRIX*

M. E. ARGENTATI†, A. V. KNYAZEV†, C. C. PAIGE‡, AND I. PANAYOTOV§

Abstract. The Rayleigh–Ritz method is widely used for eigenvalue approximation. Given a matrix X with columns that form an orthonormal basis for a subspace $\mathcal{X}$, and a Hermitian matrix A, the eigenvalues of $X^H AX$ are called Ritz values of A with respect to $\mathcal{X}$. If the subspace $\mathcal{X}$ is A-invariant, then the Ritz values are some of the eigenvalues of A. If the A-invariant subspace $\mathcal{X}$ is perturbed to give rise to another subspace $\mathcal{Y}$, then the vector of absolute values of changes in Ritz values of A represents the absolute eigenvalue approximation error using $\mathcal{Y}$. We bound the error in terms of principal angles between $\mathcal{X}$ and $\mathcal{Y}$. We capitalize on ideas from a recent paper [*SIAM J. Matrix Anal. Appl.*, 29 (2006), pp. 15–32] by Knyazev and Argentati, where the vector of absolute values of differences between Ritz values for subspaces $\mathcal{X}$ and $\mathcal{Y}$ was weakly (sub)majorized by a constant times the sine of the vector of principal angles between $\mathcal{X}$ and $\mathcal{Y}$, the constant being the spread of the spectrum of A. In that result no assumption was made on either subspace being A-invariant. It was conjectured there that if one of the trial subspaces is A-invariant, then an analogous weak majorization bound should be much stronger as it should involve only terms of the order of sine squared. Here we confirm this conjecture. Specifically we prove that the absolute eigenvalue error is weakly majorized by a constant times the sine squared of the vector of principal angles between the subspaces $\mathcal{X}$ and $\mathcal{Y}$, where the constant is proportional to the spread of the spectrum of A. For many practical cases we show that the proportionality factor is simply one and that this bound is sharp. For the general case we can prove the result only with a slightly larger constant, which we believe is artificial.

Key words. Hermitian matrices, angles between subspaces, majorization, Lidskii's eigenvalue theorem, perturbation bounds, Ritz values, Rayleigh–Ritz method, invariant subspace

AMS subject classifications. 15A18, 15A42, 15A57, 15A60

DOI. 10.1137/070684628

1. Introduction.

Eigenvalue problems appear in many applications. For example, eigenvalues represent the frequencies of vibration in mechanical vibrations, while the energy levels of a system are the eigenvalues of the Hamiltonian in quantum mechanics. Eigenvalue problems are used today in these and many other applications, including spectral data clustering and internet search engines.

Eigenvalues cannot be computed exactly except in some trivial cases, so numerical approximation is required. Eigenvalue a posteriori and a priori error bounds describe the eigenvalue approximation quality, and this is a classical and important topic in matrix analysis. A posteriori bounds are based on information readily computable, e.g., the eigenvector residuals, and are necessary, e.g., for adaptive numerical methods for eigenvalue approximation. A priori bounds are given in terms of theoretical properties and can be very useful in assessing the relative performance of algorithms.

*Received by the editors March 3, 2007; accepted for publication (in revised form) by M. Embree January 25, 2008; published electronically May 16, 2008.

http://www.siam.org/journals/simax/30-2/68462.html

†Department of Applied Mathematical and Statistical Sciences, University of Colorado Denver, P.O. Box 173364, Campus Box 170, Denver, CO (merico.argentati@cudenver.edu, andrew.knyazev@cudenver.edu). The second author's research was supported by NSF-DMS 0612751.

‡School of Computer Science, McGill University, Montreal, Quebec, H3A 2A7, Canada (paige@cs.mcgill.ca). This author's research was supported by NSERC of Canada grant OGP0009236.

§Department of Mathematics and Statistics, McGill University, Montreal, Quebec, H3A 2K6, Canada (ipanay@math.mcgill.ca). This author's research was supported by FQRNT of Quebec Scholarship 100936.

The widely used Rayleigh–Ritz method is well known for its ability to generate high quality approximations to eigenvalues of Hermitian matrices. It is the basis for many numerical procedures for computing eigenvalues, such as finite element methods and the Lanczos eigenproblem iteration. Eigenvalue error bounds for the Rayleigh–Ritz method are important, since they provide estimates and predictions of the quality of eigenvalue approximations and can be used, e.g., to predict the number of iterations needed in the Lanczos method for computing some eigenvalues to within a given accuracy. There is a vast literature on Rayleigh–Ritz eigenvalue methods and error bounds; see, e.g., [16, Chapter 4], [19, Chapters 10–13], and [20, Chapters 3–5].

We contribute to this traditional area of research with a new twist—using weak majorization. Majorization is a classical technique that can be used to formulate and prove a great variety of inequalities in a concise and elegant way. It is widely used in matrix analysis, e.g., to bound perturbations of eigenvalues via Lidskii's beautiful theorem [17]. In the context of Rayleigh–Ritz eigenvalue error bounds, weak majorization was introduced in the celebrated work of Davis and Kahan [3] to bound eigenvalue errors a posteriori. In the present paper we propose and prove what appear to be the first theorems based on weak majorization for a priori Rayleigh–Ritz eigenvalue error bounds. Our results provide a theoretical foundation that can be applied in a number of situations, e.g., for finite element methods [4] and for block Lanczos iterations such as in [5]; see [14].

We use several well known majorization results found, e.g., in [1, 7, 18]. We give references throughout the paper for the concepts we introduce. For a more thorough background and reference list, see [13].

The rest of the paper is organized as follows. Section 2 contains all necessary definitions and basic facts on majorization that we need for our eigenvalue and singular value bounds. Section 3 is the main part of the paper, where we motivate and formulate our conjectures and theorems. Section 4 has all of our proofs. In section 5 we show that our main results are sharp; we also discuss our proofs and the possibility that our bound for the most general case might be slightly improved.

2. Definitions and prerequisites. We introduce the definitions and tools we need, together with some mild motivation. We do not provide proofs for the results in this section—instead we refer the reader to some of the relevant literature.

2.1. Notation. For a real vector $x = [x_1, \ldots, x_n]^T$, we use $x^\downarrow \equiv [x_1^\downarrow, \ldots, x_n^\downarrow]^T$ to denote x with its elements rearranged in descending order, while $x^\uparrow \equiv [x_1^\uparrow, \ldots, x_n^\uparrow]^T$ denotes x with its elements rearranged in ascending order. We use $|x|$ to denote the vector x with the absolute value of its components. We use the $\leq$ symbol to compare real vectors componentwise. For real vectors x and y the expression $x \prec y$ means that x is majorized by y, while $x \prec_w y$ means that x is weakly (sub)majorized by y; see section 2.2.

We consider the Euclidean space $\mathbb{C}^n$ of column vectors equipped with the standard scalar product $x^H y$ and the norm $\|x\| = \sqrt{x^H x}$. We use the same notation $\|A\|$ for the induced matrix norm of a complex matrix $A \in \mathbb{C}^{n\times n}$. $\mathcal{X} = \mathcal{R}(X) \subset \mathbb{C}^n$ means the subspace $\mathcal{X}$ is equal to the range of the matrix X with n rows. The unit matrix is I, and the zero matrix (not necessarily square) is 0, while $e = [1, \ldots, 1]^T$. We use $\mathcal{H}(n)$ to denote the set of $n \times n$ Hermitian matrices and $\mathcal{U}(n)$ to denote the set of $n \times n$ unitary matrices in the set $\mathbb{C}^{n\times n}$ of all $n \times n$ complex matrices.

We write $\lambda(A) = \lambda^\downarrow(A)$ for the vector of eigenvalues of $A \in \mathcal{H}(n)$ arranged in descending order, and we write $s(B) = s^\downarrow(B)$ for the vector of singular values of B

arranged in descending order. Individual eigenvalues and singular values are denoted by $\lambda_i(A)$ and $s_i(B)$, respectively, so, e.g., $\mathrm{spr}(A) = \lambda_1(A) - \lambda_n(A)$ and $s_1(B) = \|B\|$.

Let subspaces $\mathcal{X}$ and $\mathcal{Y} \subseteq \mathbb{C}^n$ have the same dimension, with orthonormal bases given by the columns of the matrices X and Y, respectively. We denote the vector of principal angles between $\mathcal{X}$ and $\mathcal{Y}$ arranged in descending order by $\theta(\mathcal{X}, \mathcal{Y}) = \theta^{\downarrow}(\mathcal{X}, \mathcal{Y})$ and define it by using $\cos\theta(\mathcal{X}, \mathcal{Y}) = s^{\uparrow}(X^H Y)$, e.g., [2], [6, section 12.4.3].

2.2. Majorization and weak majorization. We now briefly define the concepts of majorization and weak majorization which are comparison relations between two real vectors. For detailed information we refer the reader to [1, 7, 18].

We say that $x \in \mathbb{R}^n$ is weakly (sub)majorized by $y \in \mathbb{R}^n$, written $x \prec_w y$, if

$$\sum_{i=1}^{k} x_i^{\downarrow} \le \sum_{i=1}^{k} y_i^{\downarrow}, \qquad 1 \le k \le n, \tag{2.1}$$

while x is (strongly) majorized by y, written $x \prec y$, if (2.1) holds together with

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i. \tag{2.2}$$

Our final results in the paper are weak majorization bounds of the form $x \prec_w y$, with $x \ge 0$. On the one hand, we can see from (2.1) that $x \le y \Rightarrow x \prec_w y$; i.e., the inequality implies weak majorization. In our case the advantage of using weak majorization is that the inequality $x \le y$ (the values of x and y become apparent later) is simply wrong, while the weak majorization bound $x \prec_w y$ does hold. On the other hand, a weak majorization bound $x \prec_w y$ implies that $\max(x) \le \max(y)$. So if the bound $\max(x) \le \max(y)$ is already known, but it is also known that $x \le y$ does not hold, it makes sense to conjecture and to try to prove $x \prec_w y$.

Strong "$\prec$" and weak "$\prec_w$" majorization relations share only some properties with the usual inequality $\le$ relation, so one should deal with them carefully. For example, $\prec$ and $\prec_w$ are reflexive and transitive, but $x \prec y$ and $y \prec x$ do not imply that $x = y$; e.g., [1, Remark II.1.2]. Similarly $x \prec y$ does not imply the intuitive $x + z \prec y + z$, as is seen in the example $x = (0, 0, 0)$, $y = (2, -1, -1)$, $z = (-2, 0, 0)$. So we must be particularly careful of the ordering when we [illegible] results. Thus it can be seen from (2.1) and (2.2) that $x + u \prec x^{\downarrow} + u^{\downarrow}$, e.g., [1, Corollary II.4.3], and

$$\{x \prec_w y\} \,\&\, \{u \prec_w v\} \,\&\, \cdots \;\Rightarrow\; x+u+\cdots \prec x^{\downarrow}+u^{\downarrow}+\cdots \prec_w y^{\downarrow}+v^{\downarrow}+\cdots, \tag{2.3}$$

where this also holds with $\prec_w$ replaced by $\prec$.

Some of the other basic majorization and related results we use are fairly obvious:

$$A \in \mathcal{H}(n) \Rightarrow |\lambda(\pm A)|^{\downarrow} = s(A); \tag{2.4}$$

$$|x \pm y| \prec_w |x|^{\downarrow} + |y|^{\downarrow}, \quad \text{since from (2.3)} \quad |x \pm y| \le |x| + |y| \prec |x|^{\downarrow} + |y|^{\downarrow}; \tag{2.5}$$

$$x \prec y \Rightarrow |x| \prec_w |y|; \quad \text{see, e.g., [1, Example II.3.5].} \tag{2.6}$$

Arithmetic operations, e.g., the sum and the product, on vectors used in majorization are performed componentwise. In the subsequent Theorems 2.3 and 2.4 for rectangular matrices we may need to operate with nonnegative vectors of different lengths. A standard agreement in this case is to add zeros at the end of the shorter vector to match the sizes needed for componentwise arithmetic operations and comparisons. We also use this agreement in later proofs.

Many inequality relations between eigenvalues and singular values are succinctly expressed as majorization or weak majorization relations; a beautiful example is the following.

THEOREM 2.1 (Lidskii [17]; see also, e.g., [1, p. 69]). *For $A, B \in \mathcal{H}(n)$ [illegible] A, B, $A - B$ [illegible] $\lambda(A) - \lambda(B) \prec \lambda(A - B)$*

Recall here that $\lambda(A) - \lambda(B) = \lambda^{\downarrow}(A) - \lambda^{\downarrow}(B)$. Note that the equivalent of (2.2) holds here by using $\operatorname{trace}(A) = \sum_i \lambda_i(A)$. We will use the following corollary.

COROLLARY 2.2 (e.g., [18, Chapter 9, G.1.d], [7, Corollary 3.4.3]). *For $A, B \in \mathbb{C}^{n\times n}$ [illegible] $s(A \pm B) \prec_w s(A) + s(B)$*

This corollary also follows from a weaker statement than Lidskii's theorem, e.g., [1, Exercises II.1.14 and II.1.15].

By using (2.3) we can see that Corollary 2.2 extends to the case of three or more matrices, because all vectors $s(A), s(B), \ldots$ are nonincreasing.

We also use results for the singular values of a product of matrices.

THEOREM 2.3 (e.g., [7, Theorem 3.3.14]). *$s(AB) \prec_w s(A)s(B)$ [illegible] A [illegible] B [illegible] AB*

THEOREM 2.4 (e.g., [7, Theorem 3.3.16], [1, Problem III.6.2]). *$s(AB) \le \|A\| s(B)$ [illegible] $s(AB) \le \|B\| s(A)$ [illegible] A [illegible] B [illegible] AB*

3. Motivation and main results. The Rayleigh–Ritz method for approximating eigenvalues of a Hermitian matrix A finds the eigenvalues of $X^H AX$, where the columns of the matrix X form an orthonormal basis for a subspace $\mathcal{X}$. Here $\mathcal{X}$ is called a trial subspace. The eigenvalues of $X^H AX$ do not depend on the particular choice of basis and are called Ritz values of A with respect to $\mathcal{X}$. If $\mathcal{X}$ is one-dimensional and spanned by the unit vector x, there is only one Ritz value—namely, the Rayleigh quotient $x^H Ax$.

When the trial subspace $\mathcal{X}$ is perturbed to become the subspace $\mathcal{Y}$, it is useful to know how the Ritz values of A vary. For one-dimensional $\mathcal{X}$ and $\mathcal{Y}$, spanned by unit vectors x and y, respectively, the following result appears in, e.g., [12, Theorem 1]:

$$|x^H Ax - y^H Ay| \le \operatorname{spr}(A) \sin\theta(x, y). \tag{3.1}$$

Here and below, $\theta(x, y)$ is the acute angle between the two unit vectors x and y defined by $\theta(x, y) = \arccos|x^H y| \in [0, \pi/2]$.

It is well known that every eigenvector is a stationary point of the Rayleigh quotient (considered as a function of a vector)—i.e., in the vicinity of an eigenvector, the Rayleigh quotient changes very slowly. The classic result that motivates this paper is the following: The Rayleigh quotient approximates an eigenvalue of a Hermitian matrix with accuracy proportional to the [illegible] of the eigenvector approximation error. The following simple bound, e.g., [12, Theorem 4], demonstrates this:

$$|x^H Ax - y^H Ay| \le \operatorname{spr}(A) \sin^2\theta(x, y), \tag{3.2}$$

where we assume that one of the unit vectors x or y is an eigenvector of A. To give a thorough background to our results, we rederive this important basic bound. Let $Ax = x\lambda$, and then $x^H Ax = \lambda$ so $|x^H Ax - y^H Ay| = |y^H (A - \lambda I)y|$. We now plug in the orthogonal decomposition $y = u + v$, where $u \in \operatorname{span}\{x\}$ and $v \in (\operatorname{span}\{x\})^{\perp}$. Thus $(A - \lambda I)u = 0$ and $\|v\| = \sin\theta(x, y)$, which results in $|y^H (A - \lambda I)y| = |v^H (A - \lambda I)v| \le \|A - \lambda I\| \|v\|^2 = \|A - \lambda I\| \sin^2\theta(x, y)$. But $\|A - \lambda I\| \le \operatorname{spr}(A)$, giving (3.2).

Let us now discuss some generalizations of (3.1) and (3.2) for subspaces $\mathcal{X}$ and $\mathcal{Y}$ of dimensions higher than one, with $\dim\mathcal{X} = \dim\mathcal{Y}$. Let X and Y be two matrices

whose columns form orthonormal bases for $\mathcal{X}$ and $\mathcal{Y}$, respectively, and suppose that the Ritz values of A with respect to $\mathcal{X}$ and $\mathcal{Y}$ are arranged in descending (more precisely "nonincreasing") order. To generalize (3.1) and (3.2) we replace the usual notion of angles between vectors by a more general one of principal angles between subspaces and replace the inequality symbol by the weak (sub)majorization symbol $\prec_w$.

Let $\lambda(A)$ denote the vector of descending eigenvalues $\lambda_i(A)$ of a Hermitian matrix A, $s(B)$ the vector of descending singular values of a matrix B, and $\theta(\mathcal{X}, \mathcal{Y})$ the vector of descending principal angles $\theta_i(\mathcal{X}, \mathcal{Y})$ between the subspaces $\mathcal{X}$ and $\mathcal{Y}$, defined such that the vectors $\cos\theta(\mathcal{X}, \mathcal{Y})$ and $s(X^H Y)$ are the same, except for the reversed order; see, e.g., [2], [6, section 12.4.3]. A recent paper [13] generalizes (3.1) to:

$$|\lambda(X^H AX) - \lambda(Y^H AY)| \prec_w \operatorname{spr}(A) \sin\theta(\mathcal{X}, \mathcal{Y}). \tag{3.3}$$

The weak majorization bound (3.3) implies, e.g., a bound for its largest term:

$$\max_i |\lambda_i(X^H AX) - \lambda_i(Y^H AY)| \leq \operatorname{spr}(A) \operatorname{gap}(\mathcal{X}, \mathcal{Y}), \tag{3.4}$$

where $\operatorname{gap}(\mathcal{X}, \mathcal{Y}) = \max_i\{\sin\theta_i(\mathcal{X}, \mathcal{Y})\}$ in this case, e.g., [11, 13].

Both bounds (3.3) and (3.4) generalize (3.1) to multidimensional subspaces, but no assumption of A-invariance is made in either case. What is the bound that generalizes (3.2), assuming that one of the subspaces $\mathcal{X}$ or $\mathcal{Y}$ is A-invariant? A natural conjecture, made in [13], is that such a bound could be obtained in terms of $\sin^2\theta(\mathcal{X}, \mathcal{Y})$. No majorization result of this kind is known, but simpler results—for the largest error only—are available; e.g., the following important bound is proved in [9] and reproduced in [4, Theorem 2, p. 477] and [15, Theorem 2.4], with a different proof suggested in [8, Theorem 2.2.3, p. 56]; for an English translation of the latter, see [10, Theorem 2.3, p. 383]. We present here a slightly modified formulation to make it consistent with (3.4): If $\mathcal{X}$ or $\mathcal{Y}$ is A-invariant and corresponds to a contiguous set of the extreme, i.e., largest or smallest, eigenvalues of A, then

$$\max_i |\lambda_i(X^H AX) - \lambda_i(Y^H AY)| \leq \operatorname{spr}(A) \operatorname{gap}^2(\mathcal{X}, \mathcal{Y}). \tag{3.5}$$

Bound (3.5) generalizes (3.2) but does not take advantage of majorization. By comparing (3.3) and (3.4) with (3.1), and (3.5) with (3.2), we make an educated guess for the general case where the invariant subspace is not necessarily associated with a contiguous set of extreme eigenvalues:

CONJECTURE 3.1. [illegible] $\mathcal{X}$ [illegible] $\mathcal{Y}$ [illegible] X [illegible] Y [illegible] A [illegible] $\mathcal{X}$ [illegible] $\mathcal{Y}$ [illegible] A [illegible]

$$|\lambda(X^H AX) - \lambda(Y^H AY)| \prec_w \operatorname{spr}(A) \sin^2\theta(\mathcal{X}, \mathcal{Y}). \tag{3.6}$$

We emphasize that the bound (3.6) involves the sine [illegible], and, since convergence analyses are of particular interest for small angles, this is a great improvement over (3.3). This is just as we would hope, since one of the subspaces is A-invariant in (3.6). The exact A-invariance assumption is equivalent to the subspace being spanned by some exact eigenvectors of A, and Conjecture 3.1 is an a priori Rayleigh–Ritz eigenvalue error bound which can be used to examine how the subspaces $\mathcal{Y}$ of an iterative eigenproblem algorithm approach an ideal A-invariant subspace $\mathcal{X}$. As we mentioned in the introduction, eigenvalue error bounds are important in many applications. We

refer the reader to the follow-up paper [14], where we extend some results of this paper to Hilbert spaces and discuss in detail applications to finite element methods and subspace iterations.

The implications of the weak majorization inequality (3.6) in Conjecture 3.1 may not be obvious to every reader. The weak majorization bound (3.6) directly implies that

$$\sum_{i=1}^{j} |\lambda_i(X^H AX) - \lambda_i(Y^H AY)|^{\downarrow} \le \mathrm{spr}(A) \sum_{i=1}^{j} \sin^2(\theta_i(\mathcal{X},\mathcal{Y}))^{\downarrow}, \quad j = 1,\dots,k;$$

see (2.1), where $k = \dim \mathcal{X} = \dim \mathcal{Y}$. For example, for $j = k$ we obtain

$$\sum_{i=1}^{k} |\lambda_i(X^H AX) - \lambda_i(Y^H AY)| \le \mathrm{spr}(A) \sum_{i=1}^{k} \sin^2(\theta_i(\mathcal{X},\mathcal{Y})),$$

and for $j = 1$ we get (3.5). Moreover, for real vectors x and y the weak majorization $x \prec_w y$ is equivalent to the inequality $\sum_{i=1}^{n} \phi(x_i) \le \sum_{i=1}^{n} \phi(y_i)$ holding for any continuous nondecreasing convex real-valued function ϕ; see, e.g., [18, Statement 4.B.2]. If, for example, we take $\phi(t) = t^p$, with $p \ge 1$, the bound (3.6) also implies that

$$\left(\sum_{i=1}^{k} |\lambda_i(X^H AX) - \lambda_i(Y^H AY)|^p \right)^{\frac{1}{p}} \le \mathrm{spr}(A) \left(\sum_{i=1}^{k} \sin^{2p}(\theta_i(\mathcal{X},\mathcal{Y})) \right)^{\frac{1}{p}}.$$

We have not proven that Conjecture 3.1 holds in [illegible] circumstances, and indeed it might not (but we suspect that it does). But we have proven that it [illegible] holds if we multiply the bound by 1.5. In section 4 we also show that Conjecture 3.1 does hold in some very useful circumstances:

THEOREM 3.1. [illegible] (3.6) [illegible] 3.1 [illegible] 3.1 [illegible]

(a) [illegible] A [illegible] $\mathcal{X}$ $\mathcal{Y}$ [illegible] A

(b) [illegible] A [illegible] A [illegible] $\mathcal{X}$ $\mathcal{Y}$ [illegible] A [illegible] $[\lambda_1(A)+\lambda_n(A)]/2$ [illegible] A [illegible]

This does not cover all known cases where (3.6) holds, but it does cover many practical cases. For example, in approximating the eigenvalues of a Hermitian matrix, perhaps by using Lanczos' eigenvalue algorithm, e.g., [6, section 9], we are often interested in just one end of the spectrum. In section 4 we also show that a weaker result [illegible] holds.

THEOREM 3.2. [illegible] 3.1 [illegible]

$$|\lambda(X^H AX) - \lambda(Y^H AY)| \prec_w \mathrm{spr}(A) \left[e - \cos\theta(\mathcal{X},\mathcal{Y}) + \frac{1}{2}\sin^2\theta(\mathcal{X},\mathcal{Y}) \right]. \tag{3.7}$$

Here and below, we use "e" to indicate a vector of ones. Note that the individual elements for both vectors $e - \cos\theta(\mathcal{X},\mathcal{Y})$ and $\sin^2\theta(\mathcal{X},\mathcal{Y})$ are decreasing, since both functions $1 - \cos\theta$ and $\sin^2\theta$ are monotonically increasing within $[0, \pi/2]$, and the vector $\theta(\mathcal{X},\mathcal{Y})$ is chosen to be decreasing. We now deduce two simple corollaries of

Theorem 3.2. By using elementary trigonometry, for $\theta \in [0, \pi/2]$:

$$\begin{aligned} 2 - 2\cos\theta &= 2 - 2\cos\theta - (1-\cos\theta)^2 + (1-\cos\theta)^2 \\ &= \sin^2\theta + (1-\cos\theta)^2 = \sin^2\theta + \sin^4\theta/(1+\cos\theta)^2 \\ &\leq \sin^2\theta + \sin^4\theta. \end{aligned}$$

We first conclude that bound (3.7) is slightly worse than bound (3.6) from Conjecture 3.1; and second, we immediately obtain from (3.7) the following.

COROLLARY 3.3. [illegible] 3.1 [illegible]

$$|\lambda(X^H AX) - \lambda(Y^H AY)| \prec_w \operatorname{spr}(A)\left[\sin^2\theta(\mathcal{X},\mathcal{Y}) + \frac{1}{2}\sin^4\theta(\mathcal{X},\mathcal{Y})\right] \tag{3.8}$$

$$\leq \frac{3}{2}\operatorname{spr}(A)\sin^2\theta(\mathcal{X},\mathcal{Y}). \tag{3.9}$$

By extending the above trigonometric relation we see that

$$2 - 2\cos\theta = \sin^2\theta\left(1 + \frac{\sin^2\theta}{(1+\cos\theta)^2}\right) = \frac{2\sin^2\theta}{1+\cos\theta} = \frac{\sin^2\theta}{\cos^2(\theta/2)} \leq \tan^2\theta$$

for $\theta \in [0, \pi/2]$; and with $\sin^2\theta \leq \tan^2\theta$, bound (3.7) implies another corollary.

COROLLARY 3.4. [illegible] 3.1 [illegible]

$$|\lambda(X^H AX) - \lambda(Y^H AY)| \prec_w \operatorname{spr}(A)\tan^2\theta(\mathcal{X},\mathcal{Y}). \tag{3.10}$$

We give an example in section 5 demonstrating that the conjectured bound (3.6) cannot be any tighter. Our numerical tests suggest that Conjecture 3.1 holds, i.e., that bound (3.7) can probably be improved to (3.6). However, we show in section 5 that already the first step in our proof of Theorem 3.2 does not allow us to prove the better bound (3.6), so a completely different approach is apparently needed to support Conjecture 3.1 in all cases—see section 5 for more thoughts on this.

Conjecture 3.1 turns out to be easy to formulate but hard to prove in its generality. We believe that the present publication, which proves Conjecture 3.1 in several practically interesting particular cases and provides slightly weaker bounds (3.7)–(3.10) for the general case, is important since it serves as a theoretical foundation for our future work on applications, e.g., [14]. It is also novel—we know of no other case where majorization is used for a priori Rayleigh–Ritz error bounds. The only somewhat related result known to us is the pioneering work of [3], where majorization is applied to bound eigenvalue errors a posteriori.

4. Proofs. We have all of the tools needed to prove our main results of Theorems 3.1 and 3.2. At first, both proofs develop along the same lines; later they split.

By the assumptions in the theorems, $\mathcal{X}$ and $\mathcal{Y}$ are two subspaces of $\mathbb{C}^n$ of the same dimension k and are the column ranges of matrices X and Y with orthonormal columns that are arbitrary up to unitary transformations of their columns. By using the singular value decomposition we choose such a pair of matrices X and Y with orthonormal columns so that $C \equiv X^H Y$ is real, square, and diagonal, with the diagonal entries in increasing order. Thus by the definition of angles between subspaces,

$$C = \operatorname{diag}\left(s^{\uparrow}(X^H Y)\right) = \operatorname{diag}\left(\cos\theta(\mathcal{X},\mathcal{Y})\right). \tag{4.1}$$

We arbitrarily complete X and Y to unitary matrices $[X, X_\perp]$ and $[Y, Y_\perp] \in \mathcal{U}(n)$, respectively, and consider the 2×2 partition of their unitary product $[X, X_\perp]^H [Y, Y_\perp]$.

By construction of X and Y, its $k \times k$ upper left block is C. We denote its $(n-k) \times k$ lower left block by $S \equiv (X_\perp)^H Y$. Since $[X, X_\perp]^H [Y, Y_\perp]$ is unitary, the entries C and S of its first block column satisfy $C^2 + S^H S = I$. So $\lambda(S^H S) = \lambda(I - C^2) = e - \cos^2\theta(\mathcal{X}, \mathcal{Y}) = \sin^2\theta(\mathcal{X}, \mathcal{Y})$, where e is the vector of ones, and so the vectors of singular values $s(C)$ and $s(S)$ are closely connected, and we derive from this that

$$\sin\theta(\mathcal{X}, \mathcal{Y}) = [s(S), 0, \ldots, 0], \tag{4.2}$$

where $\max\{2k-n, 0\}$ zeros are added on the right-hand side to match the number k of angles in the vector $\theta(\mathcal{X}, \mathcal{Y})$ with the number $\min\{k, n-k\}$ of singular values in the vector $s(S)$.

Both theorems assume that either $\mathcal{X}$ or $\mathcal{Y}$ is A-invariant, so without loss of generality let $\mathcal{X}$ be A-invariant. Then since $[X, X_\perp]$ is unitary:

$$[X, X_\perp]^H A\, [X, X_\perp] = \operatorname{diag}(A_{11}, A_{22}) \text{ and } A = [X, X_\perp] \operatorname{diag}(A_{11}, A_{22}) [X, X_\perp]^H.$$

Here $X^H A X = A_{11} \in \mathcal{H}(k)$ and $(X_\perp)^H A X_\perp = A_{22} \in \mathcal{H}(n-k)$. We can now use $Y^H [X, X_\perp] = [C^H, S^H] = [C, S^H]$ to show that

$$Y^H A Y = Y^H \left([X, X_\perp] \operatorname{diag}(A_{11}, A_{22}) [X, X_\perp]^H \right) Y = C A_{11} C + S^H A_{22} S. \tag{4.3}$$

The expression we want to bound in Theorems 3.1 and 3.2 now takes the form

$$\begin{aligned} \lambda(X^H A X) - \lambda(Y^H A Y) &= \lambda(A_{11}) - \lambda(C A_{11} C + S^H A_{22} S) \\ &= \lambda(A_{11}) - \lambda(C A_{11} C) + \lambda(C A_{11} C) - \lambda(C A_{11} C + S^H A_{22} S) \\ &\prec [\lambda(A_{11}) - \lambda(C A_{11} C)]^{\downarrow} + \lambda(-S^H A_{22} S), \end{aligned} \tag{4.4}$$

where the last line used Lidskii's Theorem 2.1 with (2.3). See the discussion following (5.1) for more about this choice. Next (2.4), Theorems 2.3 and 2.4, and (4.2) give

$$\left| \lambda(-S^H A_{22} S) \right|^{\downarrow} = s(S^H A_{22} S) \prec_w \|A_{22}\| \sin^2\theta(\mathcal{X}, \mathcal{Y}). \tag{4.5}$$

At this point, the proofs split. Each proof will use a different majorization of $\lambda(A_{11}) - \lambda(C A_{11} C)$ in (4.4), but both will use (4.5). We first establish Theorem 3.1. Neither (3.6) nor (3.7) is altered by replacing A by $\pm A + \alpha I$, where α is an arbitrary real constant, and so we can make the new A_{11} nonnegative definite in each of the parts (a) and (b) of Theorem 3.1 by choosing the appropriate sign and the shift α.

[illegible] 3.1. The starting point of the proof is (4.4), but now we assume that A_{11} is nonnegative definite and so has a nonnegative definite square root $\sqrt{A_{11}}$. We deal with $\lambda(A_{11}) - \lambda(C A_{11} C)$ first. For arbitrary square matrices F and G, we have $\lambda(FG) = \lambda(GF)$. By taking $F = C\sqrt{A_{11}}$ and $G = \sqrt{A_{11}} C$, we get $\lambda(C A_{11} C) = \lambda(\sqrt{A_{11}} C^2 \sqrt{A_{11}})$. By using this and Lidskii's Theorem 2.1, we see that

$$\begin{aligned} \lambda(A_{11}) - \lambda(C A_{11} C) &= \lambda(A_{11}) - \lambda\left(\sqrt{A_{11}} C^2 \sqrt{A_{11}} \right) \\ &\prec \lambda\left(\sqrt{A_{11}} \sqrt{A_{11}} - \sqrt{A_{11}} C^2 \sqrt{A_{11}} \right) \\ &= \lambda\left(\sqrt{A_{11}} \left(I - C^2\right) \sqrt{A_{11}} \right) = \lambda\left(\sqrt{A_{11}} S^H S \sqrt{A_{11}} \right), \end{aligned}$$

since $C^2 + S^H S = I$. Then by using (2.6) with Theorem 2.4 (twice) and (4.2), we obtain

$$|\lambda(A_{11}) - \lambda(C A_{11} C)| \prec_w s\left(\sqrt{A_{11}} S^H S \sqrt{A_{11}} \right) \le \|A_{11}\| \sin^2\theta(\mathcal{X}, \mathcal{Y}).$$

Apply (2.6) to (4.4); then (2.5), (2.3), and (4.5) with the above bound give

$$\begin{aligned}\left|\lambda(X^HAX) - \lambda(Y^HAY)\right| &\prec_w \left|[\lambda(A_{11}) - \lambda(CA_{11}C)]^{\downarrow} + \lambda(-S^HA_{22}S)\right| \\ &\prec_w |\lambda(A_{11}) - \lambda(CA_{11}C)|^{\downarrow} + \left|\lambda(-S^HA_{22}S)\right|^{\downarrow} \\ &\prec_w (\|A_{11}\| + \|A_{22}\|)\sin^2\theta(\mathcal{X},\mathcal{Y}). \qquad (4.6)\end{aligned}$$

Here this proof splits, and we first prove part (a) of Theorem 3.1. By assumption the invariant subspace $\mathcal{X}$ corresponds to a contiguous set of the largest (or smallest) eigenvalues of A. Here we present the proof for the case of the largest eigenvalues. The case of the smallest eigenvalues follows immediately by substituting $-A$ for A. We replace A with $A + \alpha I$, where α is chosen as the constant real shift that makes the new A_{11} positive semidefinite (nonnegative definite and singular), so that $\sqrt{A_{11}}$ exists. Since $\dim \mathcal{X} = k$, and the invariant subspace $\mathcal{X}$ corresponds to a contiguous set of the largest eigenvalues of A, $\alpha = -\lambda_k(A)$. After the shift $\lambda_k(A)$ becomes zero, the eigenvalues of the block A_{11} become nonnegative, with $\lambda_1(A)$ being the largest in absolute value, and the eigenvalues of the block A_{22} become nonpositive, with $\|A_{22}\| = -\lambda_n(A)$. Thus $\|A_{11}\| + \|A_{22}\| = \lambda_1(A) - \lambda_n(A) = \mathrm{spr}(A)$. Using this together with (4.6) gives (3.6), completing the proof of part (a).

For part (b) of Theorem 3.1 we prove the case where the eigenvalues of A_{11} lie in the top half of the spectrum of A; the remaining case is proven by substituting $-A$ for A. Choose the shift so that, for the new A, $\lambda_1(A) = -\lambda_n(A)$, ensuring with the assumptions that A_{11} is nonnegative definite and that $\|A_{11}\| \le \mathrm{spr}(A)/2$ and $\|A_{22}\| \le \mathrm{spr}(A)/2$, so that (4.6) again leads to (3.6). ∎

In fact whenever we can choose the sign and shift in $\pm A + \alpha I$ so that this new A has A_{11} nonnegative definite with $\|A_{11}\| + \|A_{22}\| \le \mathrm{spr}(A)$, then (3.6) will be satisfied.

We return again to (4.4) and (4.5) to establish Theorem 3.2.

Proof of Theorem 3.2. Applying Lidskii's Theorem 2.1 with (2.3) to (4.4) gives

$$\begin{aligned}\lambda(X^HAX) - \lambda(Y^HAY) &\prec [\lambda(A_{11}) - \lambda(CA_{11}C)]^{\downarrow} + \lambda(-S^HA_{22}S) \\ &\prec \lambda(A_{11} - CA_{11}C) + \lambda(-S^HA_{22}S). \qquad (4.7)\end{aligned}$$

In order to bound this we will use the identity

$$A_{11} - CA_{11}C = (I-C)A_{11} + CA_{11}(I-C), \qquad (4.8)$$

together with the following results obtained by using (4.1) with Theorems 2.4 and 2.3:

$$s((I-C)A_{11}) \le \|A_{11}\|s(I-C) = \|A_{11}\|(e - \cos\theta(\mathcal{X},\mathcal{Y})), \qquad (4.9)$$

$$\begin{aligned}s(CA_{11}(I-C)) &\prec_w s(C)s(A_{11}(I-C)) \le s(A_{11}(I-C)) \\ &\le \|A_{11}\|s(I-C) = \|A_{11}\|(e - \cos\theta(\mathcal{X},\mathcal{Y})). \qquad (4.10)\end{aligned}$$

Discarding the first C in $s(CA_{11}(I-C))$ is no real loss; see section 5. Using (4.8) and applying (2.4), Corollary 2.2, and (2.3) with (4.9) and (4.10) gives

$$\begin{aligned}|\lambda(A_{11} - CA_{11}C)|^{\downarrow} &= s((I-C)A_{11} + CA_{11}(I-C)) \\ &\prec_w s((I-C)A_{11}) + s(CA_{11}(I-C)) \\ &\prec_w 2\|A_{11}\|(e - \cos\theta(\mathcal{X},\mathcal{Y})). \qquad (4.11)\end{aligned}$$

Now apply (2.6) to (4.7), followed by (2.5), and use (4.11) and (4.5) with (2.3), together with $\|A_{11}\|, \|A_{22}\| \le \|A\|$, to obtain:

$$\begin{aligned} \left|\lambda(X^HAX) - \lambda(Y^HAY)\right| &\prec_w \left|\lambda(A_{11} - CA_{11}C) + \lambda(-S^HA_{22}S)\right| \\ &\prec_w \left|\lambda(A_{11} - CA_{11}C)\right|^{\downarrow} + \left|\lambda(-S^HA_{22}S)\right|^{\downarrow} \\ &\prec_w \|A\| \left[2(e - \cos\theta(\mathcal{X},\mathcal{Y})) + \sin^2\theta(\mathcal{X},\mathcal{Y})\right]. \end{aligned} \tag{4.12}$$

Our final step is to replace $\|A\|$ by an expression involving $\mathrm{spr}(A)$. Observe here that the difference between Ritz values is invariant under shift $\alpha \in \mathbb{R}$. So we shift A in a way to minimize $\|A\|$. This situation occurs when 0 is exactly in the middle of the spectrum, in which case $\|A\| = \mathrm{spr}(A)/2$. Combining this observation with (4.12) completes the proof of (3.7). ∎

5. Discussion. The following example shows that the conjectured bound (3.6) cannot be improved as a general result. Let $n = 2m$, and let an arbitrary set of m angles θ_i be given, where $\pi/2 \ge \theta_1 \ge \cdots \ge \theta_m \ge 0$. Let $C = \mathrm{diag}(\cos(\theta_1), \ldots, \cos(\theta_m))$, $X = [I, 0]^H$, $Y = \left[C, \sqrt{I - C^2}\right]^H$, and $A = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$, where all unit matrices I are of size m, so that X and Y are $n \times m$ and A is $n \times n$. Then the θ_i become the principal angles between the pair of $k = m$ dimensional subspaces $\mathcal{X} \equiv \mathcal{R}(X)$ and $\mathcal{Y} \equiv \mathcal{R}(Y)$. Moreover the Ritz values are the eigenvalues of $X^HAX = I$ and $Y^HAY = 2C^2 - I$, and so $\left|\lambda(X^HAX) - \lambda(Y^HAY)\right|^{\downarrow} = 2\sin^2\theta(\mathcal{X},\mathcal{Y})$. In this example $\mathrm{spr}(A) = 1 - (-1) = 2$, so (3.6) turns into an equality.

Asymptotically where all of the angles are small, bounds (3.6), (3.7), (3.8), and (3.10) are all equivalent. Moreover our numerical tests support Conjecture 3.1 in all cases. Perhaps in practical terms, from the point of view of a numerical analyst, we are done. However, it would be pleasing to know whether Conjecture 3.1 holds theoretically in its generality, since bound (3.6) looks more aesthetic and cannot be improved as a general result.

One important thing we know is that our approach of starting with Theorem 2.1 to deduce (4.4) (used in the proof of (3.7)) cannot reduce bound (3.7) to bound (3.6) in general, no matter how we modify the rest of the proof. This can be seen from the following example in $\mathbb{C}^4$. Let $A = \mathrm{diag}(A_{11}, A_{22})$ and $C = X^HY$, $S = X_\perp^HY$ be as in

$$A = \left[\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right], \quad [X, X_\perp] = I_4, \quad [Y, Y_\perp] = \left[\begin{array}{cc|cc} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right], \tag{5.1}$$

where I_4 is the 4×4 unit matrix, so that $X, X_\perp, Y, Y_\perp$ all have two columns. Then $[X, X_\perp]^H[Y, Y_\perp] = [Y, Y_\perp]$ are chosen as in our proofs, and we see that $\theta(\mathcal{X},\mathcal{Y}) = [\pi/2, 0]^T$, $CA_{11}C = 0$, $S^HA_{22}S = \mathrm{diag}(1, 0)$. Here the largest and smallest eigenvalues of A are ± 1, so $\mathrm{spr}(A) = 2$. Hence by direct calculation

$$X^HAX = A_{11} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y^HAY = CA_{11}C + S^HA_{22}S = S^HA_{22}S = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

$$\left|\lambda(X^HAX) - \lambda(Y^HAY)\right| = \left|\begin{bmatrix} 1 \\ -1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right| = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \prec_w \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \mathrm{spr}(A)\sin^2\theta(\mathcal{X},\mathcal{Y}),$$

so example (5.1) satisfy (3.6).

Let us now attempt to use (4.4) for (5.1). The right-hand side of (4.4) is

$$a \equiv [\lambda(A_{11}) - \lambda(CA_{11}C)]^{\downarrow} + \lambda(-S^H A_{22} S) = \begin{bmatrix} 1 \\ -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix},$$

where it is true that $|a| \prec_w \operatorname{spr}(A) \sin^2 \theta(\mathcal{X}, \mathcal{Y})$. That is, the absolute value of the right-hand side of (4.4) is not always weakly majorized by $\operatorname{spr}(A) \sin^2 \theta(\mathcal{X}, \mathcal{Y})$, so we cannot obtain a general proof of (3.6) by starting from the majorization in (4.4).

Example (5.1) can tell us even more. For matrix $M = M^H$ we have the following generalization of (4.7) ($M = CA_{11}C$ in (4.4) and (4.7)):

$$\begin{aligned} \lambda(X^H AX) - \lambda(Y^H AY) &= \lambda(X^H AX) - \lambda(M) + \lambda(M) - \lambda(Y^H AY) \\ &\prec \lambda(X^H AX - M) + \lambda(M - Y^H AY) \equiv \tilde{a}. \end{aligned}$$

It might be thought that if, e.g., $X^H AX$ is indefinite, some such M could be chosen to minimize $\tilde{a}$ and to prove (3.6). But in example (5.1) it can be shown that there is real symmetric M giving $\tilde{a}$ satisfying the desired bound $|\tilde{a}| \prec_w \operatorname{spr}(A) \sin^2 \theta(\mathcal{X}, \mathcal{Y})$. In particular $M = Y^H AY$ will not give this bound, as the reader can check via (5.1). That is, using $\lambda(X^H AX) - \lambda(Y^H AY) \prec \lambda(A_{11} - CA_{11}C - S^H A_{22} S)$ in place of (4.4) will still not give (3.6) via our approach.

So, on the one hand, we cannot improve bound (3.7) to give (3.6) except possibly by considering a different approach to our present way of using Lidskii's Theorem 2.1 or equivalent in the first step; see (4.4) and (4.7). On the other hand, our numerical tests suggest that the tighter bound (3.6) holds. Thus if we are to prove (3.6) for widely spread interior eigenvalues, we appear to need an approach more sophisticated than our particular application of Lidskii's theorem in the first step.

An essentially equivalent first step was used in [12, Theorem 10] in an earlier attempt to prove (3.3), where it led to an artificial multiplier $\sqrt{2}$ in the right-hand side of (3.3). The subsequent paper [13] used an unusual technique to extend an arbitrary Hermitian operator to an orthogonal projector in a higher-dimensional space, preserving its Ritz values, to prove (3.3) as it is stated, without the multiplier $\sqrt{2}$. Perhaps the same technique might shed light here and help us to establish Conjecture 3.1, but this currently remains an open question.

6. Conclusions. We clarify a conjecture of Knyazev and Argentati [13] on a bound for the absolute difference between Ritz values of a Hermitian matrix A for two trial subspaces, one of which is A-invariant. We prove the conjecture for the cases where (a) the A-invariant subspace corresponds to a contiguous set of the largest (or smallest) eigenvalues of A and (b) the eigenvalues of A corresponding to the A-invariant subspace all lie in the top (or the bottom) half of the spectrum of A. We prove a slightly weaker bound for general invariant subspaces. We believe that the conjecture holds, i.e., that this weaker bound can be improved, and this is supported by our numerical tests, but the proof of the conjecture in its generality (if it is true) may require an unorthodox approach, perhaps one such as that used in [13]. These results are useful in practice and, for example, are applicable to the analysis of routines which use the Rayleigh–Ritz method, such as some Krylov subspace methods. We refer the reader to the subsequent paper [14], where we extend some results of this paper to Hilbert spaces and discuss in detail their application to finite element methods and subspace iterations.

REFERENCES

[1] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.

[2] Å. Björck and G. H. Golub, *Numerical methods for computing angles between linear subspaces*, Math. Comp., 27 (1973), pp. 579–594.

[3] C. Davis and W. M. Kahan, *The rotation of eigenvectors by a perturbation.* III, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.

[4] E. G. D'yakonov, *Optimization in Solving Elliptic Problems*, CRC Press, Boca Raton, FL, 1996.

[5] G. H. Golub and R. Underwood, *The Block Lanczos Method for Computing Eigenvalues*, in Mathematical Software III, J. Rice, ed., Academic Press, New York, 1977, pp. 364–377.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1996.

[7] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.

[8] A. V. Knyazev, *Computation of Eigenvalues and Eigenvectors for Mesh Problems: Algorithms and Error Estimates*, manuscript, Department of Numerical Mathematics, USSR Academy of Sciences, Moscow, 1986 (in Russian).

[9] A. V. Knyazev, *Sharp a priori error estimates of the Rayleigh-Ritz method without assumptions of fixed sign or compactness*, Math. Notes, 38 (1986), pp. 998–1002.

[10] A. V. Knyazev, *Convergence rate estimates for iterative methods for mesh symmetric eigenvalue problem*, Soviet J. Numer. Anal. Math. Modelling, 2 (1987), pp. 371–396.

[11] A. V. Knyazev and M. E. Argentati, *Principal angles between subspaces in an A-based scalar product: Algorithms and perturbation estimates*, SIAM J. Sci. Comput., 23 (2002), pp. 2008–2040.

[12] A. V. Knyazev and M. E. Argentati, *On proximity of Rayleigh quotients for different vectors and Ritz values generated by different trial subspaces*, Linear Algebra Appl., 415 (2006), pp. 82–95.

[13] A. V. Knyazev and M. E. Argentati, *Majorization for changes in angles between subspaces, Ritz values, and graph Laplacian spectra*, SIAM J. Matrix Anal. Appl., 29 (2006), pp. 15–32.

[14] A. V. Knyazev and M. E. Argentati, *Rayleigh-Ritz Majorization Error Bounds with Applications to FEM and Subspace Iterations*, http://arxiv.org/abs/math/0701784.

[15] A. V. Knyazev and J. E. Osborn, *New a priori FEM error estimates for eigenvalues*, SIAM J. Numer. Anal., 43 (2006), pp. 2647–2667.

[16] M. A. Krasnosel'skii, G. M. Vainikko, P. P. Zabreiko, Ya. B. Rutitskii, and Y. Ya. Stetsenko, *Approximate Solutions of Operator Equations*, Wolters-Noordhoff, Groningen, 1972, (in English).

[17] V. B. Lidskii, *On the proper values of a sum and product of symmetric matrices*, Dokl. Akad. Nauk, 75 (1950), pp. 769–772.

[18] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and its Applications*, Academic Press, New York, 1979.

[19] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, PA, 1998.

[20] H. F. Weinberger, *Variational Methods for Eigenvalue Approximation*, SIAM, Philadelphia, PA, 1974.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 560–581

EIGENVALUES OF THE SUM OF MATRICES FROM UNITARY SIMILARITY ORBITS*

CHI-KWONG LI†, YIU-TUNG POON‡, AND NUNG-SING SZE§

Abstract. Let A and B be $n \times n$ complex matrices. Characterization is given for the set $\mathcal{E}(A, B)$ of eigenvalues of matrices of the form $U^*AU + V^*BV$ for some unitary matrices U and V. Consequences of the results are discussed and computer algorithms and programs are designed to generate the set $\mathcal{E}(A, B)$. The results refine those of Wielandt on normal matrices. Extensions of the results to the sum of matrices from three or more unitary similarity orbits are also considered.

Key words. eigenvalues, sum of matrices, unitary similarity orbits, Davis–Wielandt shell

AMS subject classification. 15A18

DOI. 10.1137/070699123

1. Introduction. Denote by M_n the set of $n \times n$ complex matrices. Let $A, B \in M_n$. There has been a great deal of interest in studying the eigenvalues of matrices of the form $U^*AU + V^*BV$ for some unitary matrices $U, V \in M_n$ because of motivations from theory as well as applications; see [1, 2, 4, 7, 11, 18, 19]. The study has been very successful for Hermitian matrices. Klyachko [12] (see also [9, 11, 13], etc.) gave necessary and sufficient conditions for the real numbers $c_1, \ldots, c_n$ to be the eigenvalues of the sum of two Hermitian matrices in M_n with eigenvalues $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$.

The problem for non-Hermitian matrices is more challenging. For two given matrices $A, B \in M_n$, let $\mathcal{E}(A, B)$ be the set of eigenvalues of matrices of the form $U^*AU + V^*BV$ for some unitary matrices U and V. Wielandt [20] (see also [3] and [16]) determined the set $\mathcal{E}(A, B)$ for two normal matrices $A, B \in M_n$. There is not much information about the set $\mathcal{E}(A, B)$ for general matrices $A, B \in M_n$. The purpose of this paper is to address this problem.

In section 2, we characterize $\mathcal{E}(A, B)$ for two given matrices $A, B \in M_n$. Additional results concerning normal matrices and essentially Hermitian matrices (normal matrices with collinear eigenvalues) are presented in sections 3 and 4. In section 5, we consider an extension of our results to the sum of three of more matrices and mention some related problems. In section 6, we describe how to use our results to design computer algorithms and programs to generate the set $\mathcal{E}(A, B)$.

2. Main results. First, we characterize the matrix pair $(A, B) \in M_n \times M_n$ such that $0 \notin \mathcal{E}(A, B)$. We need the concept of Davis–Wielandt shell [5, 6] of $A \in M_n$ defined by

$$DW(A) = \{(x^*Ax, x^*A^*Ax) : x \in \mathbb{C}^n, x^*x = 1\} \subseteq \mathbb{C} \times \mathbb{R} \sim \mathbb{R}^3.$$

*Received by the editors August 2, 2007; accepted for publication by M. Chu January 2, 2008; published electronically June 6, 2008.
http://www.siam.org/journals/simax/30-2/69912.html

†Department of Mathematics, College of William and Mary, Williamsburg, VA 23185 (ckli@math.wm.edu). This author's research was supported by a USA NSF grant and an HK RCG grant.

‡Department of Mathematics, Iowa State University, Ames, IA 50011 (ytpoon@iastate.edu).

§Department of Mathematics, University of Connecticut, Storrs, CT 06269 (sze@math.uconn.edu). This author's research was supported by an HK RCG grant.

THEOREM 2.1. [illegible] $A, B \in M_n$ [illegible]
(a) $\det(U^*AU + V^*BV) \neq 0$ [illegible] $U, V \in M_n$
(b) $DW(A) \cap DW(-B) = \emptyset$
(c) [illegible] $\xi \in \mathbb{C}$ [illegible] $A+\xi I_n$ [illegible] $B-\xi I_n$ [illegible] $[0, \infty)$

[illegible]. If (c) holds, then $\|(A+\xi I_n)u\| > \|(B-\xi I_n)v\|$ for all unit vectors $u,\ v \in \mathbb{C}^n$, or $\|(A+\xi I_n)u\| < \|(B-\xi I_n)v\|$ for all unit vectors $u,\ v \in \mathbb{C}^n$. Thus, $(U^*AU + V^*BV)x \neq 0$ for all unitary matrices $U,\ V$ and unit vector $x \in \mathbb{C}^n$. Hence, condition (a) holds.

Suppose (a) holds. Assume that $DW(A) \cap DW(-B)$ is nonempty. Then there are orthonormal pairs (u_1, u_2) and (v_1, v_2) such that

$$Au_1 = \mu u_1 + \nu u_2 \quad \text{and} \quad -Bv_1 = \mu v_1 + \nu v_2$$

with $(\mu, \mu^2+\nu^2) \in DW(A) \cap DW(-B)$. Suppose U is unitary with u_1, u_2 as its first two columns, and V is unitary with v_1, v_2 as its first two columns. Then $U^*AU + V^*BV$ has zero first column, and hence has zero determinant, which is a contradiction. So, (b) holds.

Suppose (b) holds. Assume $n \geq 3$. Since $DW(A)$ and $DW(-B)$ are compact convex sets, by the separation theorem, there is a linear functional f such that $f(\alpha) > f(\beta)$ for all $(\alpha, \beta) \in DW(A) \times DW(-B)$. So, there is $\nu \in \mathbb{R}$ and $\mu \in \mathbb{C}$ such that

$$x^*(\nu A^*A + \mu A + \overline{\mu} A^*)x > y^*(\nu B^*B - \mu B - \overline{\mu} B^*)y$$

for any unit vectors $x, y \in \mathbb{C}^n$. We may perturb ν and assume that $\nu \neq 0$. Furthermore, we assume that $\nu > 0$; otherwise, multiply -1 to the inequality. Then for $\xi = \bar{\mu}/\sqrt{\nu}$, we see that

$$x^*(A + \xi I_n)^*(A + \xi I_n)x > y^*(B - \xi I_n)^*(B - \xi I_n)y$$

for all unit vectors $x, y \in \mathbb{C}^n$. So, condition (c) holds. □

Assume n = 2. Let $S = \{(\mu, r) \in \mathbb{R} \times \mathbb{C},\ |\mu|^2 \leq r\}$ and ∂S denote the boundary of S. By Theorem 2.1(b) and Theorem 2.2 in [15], $DW(A)$ is an ellipsoid in S and $(\mu, |\mu|^2) \in DW(A) \cap \partial S$ for every eigenvalue μ of A. Similarly, $DW(-B)$ is an ellipsoid in S and $(\tilde{\mu}, |\tilde{\mu}|^2) \in DW(-B) \cap \partial S$ for every eigenvalue $\tilde{\mu}$ of $-B$. Since $DW(A) \cap DW(-B) = \emptyset$, we see that $DW(-B)$ cannot lie in the interior of conv $DW(A)$, the convex hull of $DW(A)$. Otherwise, $DW(-B) \cap \mathcal{S} = \emptyset$. Similarly, $DW(A)$ cannot lie in the interior of conv $DW(-B)$. So, conv $DW(A) \cap$ conv $DW(-B) = \emptyset$. We can apply the argument for the cases when $n \geq 3$ to show that condition (c) holds.

Note that $\mu \in \mathcal{E}(A, B)$ if and only if there exist unitary matrices $U, V \in M_n$ such that $\det(UAU^* + VBV^* - \mu I_n) = 0$. Using Theorem 2.1, we have the following.

THEOREM 2.2. [illegible] $A, B \in M_n$ [illegible] $\mu \in \mathbb{C}$ [illegible]
(a) $\mu \notin \mathcal{E}(A, B)$
(b) $DW(A) \cap DW(\mu I_n - B) = \emptyset$
(c) [illegible] $\xi \in \mathbb{C}$ [illegible] $A+\xi I_n$ [illegible] $B-\mu I_n-\xi I_n$ [illegible] $[0, \infty)$

3. Normal matrices. If $A, B \in M_n$ are normal, then $DW(A)$ and $DW(\mu I_n - B)$ are polytopes with at most n vertices in $\mathbb{C} \times \mathbb{R} \sim \mathbb{R}^3$. We have the following.

THEOREM 3.1. [illegible] $A, B \in M_n$ [illegible] (a)–(c) [illegible] 2.2 [illegible]

(d) [illegible] A, $\mu I_n - B$ [illegible].

Proof. Suppose A and B are normal. Then the singular values of A and $\mu I_n - B$ are the absolute values of the eigenvalues of the two matrices. One readily sees that Theorem 2.2(c) is equivalent to condition (d). □

Theorem 3.1 has been proven by Wielandt [20, Theorem 1], where both lines and circles are used for the separation. As pointed out in [20], $\mathcal{E}(A, B)$ depends only on the spectra $\sigma(A)$, $\sigma(B)$ of A and B. Hence, for any nonempty finite subsets S, T of $\mathbb{C}$, we can define $\mathcal{E}(S, T) = \mathcal{E}(A, B)$, where A and B are any normal matrices of the same size such that $\sigma(A) = S$ and $\sigma(B) = T$.

If each of A and B has at most two distinct eigenvalues, then $\mathcal{E}(A, B)$ can be easily determined by Theorem 4.6 in section 4. For other cases, we have the following theorem, which is useful in constructing the set $\mathcal{E}(A, B)$ analytically or using computer programs; see section 6.

THEOREM 3.2. [illegible] $A, B \in M_n$ [illegible] 3 [illegible] 2 [illegible] (a)–(c) [illegible] 2.2 [illegible]

(e) [illegible] $(p, q) \in \{(2, 3), (3, 2)\}$ [illegible] p [illegible] A [illegible] q [illegible] B [illegible]

[illegible]

$$\mathcal{E}(A, B) = \bigcup \{\mathcal{E}(S, T) : S \subseteq \sigma(A), T \subseteq \sigma(B) \text{ [illegible] } (|S|, |T|) \in \{(2, 3), (3, 2)\}\},$$

[illegible] $|S|$ [illegible] $|T|$ [illegible] S [illegible] T [illegible].

Proof. Suppose A or B has at least 3 distinct eigenvalues and the other has at least 2 distinct eigenvalues. Then condition (d) fails to hold if and only if there are p distinct eigenvalues of A and q distinct eigenvalues of B with $(p, q) \in \{(3, 2), (2, 3)\}$ constituting an obstacle for the existence of the circle [14, Theorem 8.2]. Thus, Theorem 3.1(d) is equivalent to (e). □

To construct $\mathcal{E}(A, B)$, one can further reduce the collection of subsets in the above theorem. To this end, we need the following lemma showing that there is a one-one correspondence between the triangles on the boundary faces of the convex set $DW(B)$ and those on the boundary faces of $DW(\mu I - B)$ with $\mu = s + it$.

LEMMA 3.3. [illegible] $s, t, a_j, b_j \in \mathbb{R}$, $1 \le j \le 5$ [illegible]

$$P_j = \left(a_j, b_j, a_j^2 + b_j^2\right) \text{ [illegible] } Q_j = \left(s - a_j, t - b_j, (s - a_j)^2 + (t - b_j)^2\right).$$

[illegible] P_1, P_2, P_3 [illegible] P_4 [illegible] P_5 [illegible] ([illegible]) [illegible] P_1, P_2, P_3 [illegible] Q_4 [illegible] Q_5 [illegible] ([illegible]) [illegible] Q_1, Q_2, Q_3 [illegible].

Proof. Suppose P_1, P_2, P_3 are not collinear. Then Q_1, Q_2, Q_3 are not collinear. Let Π_1 and Π_2 be the planes determined by P_1, P_2, P_3 and Q_1, Q_2, Q_3, respectively.

For $(a_{pq}) \in M_3$, denote by $\det((a_{pq})) = |a_{pq}|$. For $j = 4, 5$, we have

$$((P_2-P_1)\times(P_3-P_1))\cdot(P_j-P_1)=\begin{vmatrix} a_2-a_1 & b_2-b_1 & a_2^2+b_2^2-a_1^2-b_1^2 \\ a_3-a_1 & b_3-b_1 & a_3^2+b_3^2-a_1^2-b_1^2 \\ a_j-a_1 & b_j-b_1 & a_j^2+b_j^2-a_1^2-b_1^2 \end{vmatrix}$$

and

$$\begin{aligned} & ((Q_2-Q_1)\times(Q_3-Q_1))\cdot(Q_j-Q_1) \\ = & \begin{vmatrix} a_1-a_2 & b_1-b_2 & a_2^2+b_2^2-a_1^2-b_1^2+2s(a_1-a_2)+2t(b_1-b_2) \\ a_1-a_3 & b_1-b_3 & a_3^2+b_3^2-a_1^2-b_1^2+2s(a_1-a_3)+2t(b_1-b_3) \\ a_1-a_j & b_1-b_j & a_j^2+b_j^2-a_1^2-b_1^2+2s(a_1-a_j)+2t(b_1-b_j) \end{vmatrix} \\ = & \begin{vmatrix} a_1-a_2 & b_1-b_2 & a_2^2+b_2^2-a_1^2-b_1^2 \\ a_1-a_3 & b_1-b_3 & a_3^2+b_3^2-a_1^2-b_1^2 \\ a_1-a_j & b_1-b_j & a_j^2+b_j^2-a_1^2-b_1^2 \end{vmatrix} \\ = & ((P_2-P_1)\times(P_3-P_1))\cdot(P_j-P_1). \end{aligned}$$

The result follows from the fact that P_4 and P_5 lie in the same open half space determined by Π_1 if and only if the triple products

$$((P_2-P_1)\times(P_3-P_1))\cdot(P_4-P_1) \quad \text{and} \quad ((P_2-P_1)\times(P_3-P_1))\cdot(P_5-P_1)$$

have the same sign and similar assertion for Q_j and Π_2. □

THEOREM 3.4. [illegible] $A, B \in M_n$ [illegible] $a_1, \ldots, a_n$ [illegible] $b_1, \ldots, b_n$ [illegible] $\mu \in \mathcal{E}(A,B)$ [illegible] $X = \operatorname{diag}(w_1, w_2, w_3)$ [illegible] $Y = \operatorname{diag}(z_1, z_2)$ [illegible] $DW(X) \cap DW(\mu I_2 - Y) \neq \emptyset$ [illegible]

(a) $w_1, w_2, w_3 \in \sigma(A)$ [illegible] $z_1, z_2 \in \sigma(B)$ [illegible] $DW(\operatorname{diag}(w_1, w_2, w_3))$ [illegible] $DW(A)$ [illegible] $DW(\operatorname{diag}(z_1, z_2))$ [illegible] $DW(B)$.

(b) $w_1, w_2, w_3 \in \sigma(B)$ [illegible] $z_1, z_2 \in \sigma(A)$ [illegible] $DW(\operatorname{diag}(w_1, w_2, w_3))$ [illegible] $DW(B)$ [illegible] $DW(\operatorname{diag}(z_1, z_2))$ [illegible] $DW(A)$.

[illegible]. Note that for any $z_1, z_2, z_3 \in \sigma(B)$, $DW(\operatorname{diag}(\mu - z_1, \mu - z_2, \mu - z_3))$ lies on the boundary of $DW(\mu I_n - B)$ if and only if $DW(\operatorname{diag}(z_1, z_2, z_3))$ lies on the boundary of $DW(B)$. Now, $DW(A)$ and $DW(\mu I_n - B)$ are two convex polytopes in $\mathbb{C} \times \mathbb{R}$ with vertices in $\mathbf{P} = \{(z, |z|^2) : z \in \mathbb{C}\}$. So, $DW(A) \cap DW(\mu I_n - B) \neq \emptyset$ if and only if one of the polytopes intersects a boundary face of the other polytopes. Suppose $DW(\mu I_n - B)$ intersects a boundary face of $DW(A)$. Then there are three vertices, say $(w_j, |w_j|^2)$ with $w_j \in \sigma(A)$ for $j = 1, 2, 3$, of the boundary face of $DW(A)$ intersecting $DW(\mu I_n - B)$. Note that the vertices of $DW(\mu I_n - B)$ belong to $\mathbf{P}$. So, $DW(\operatorname{diag}(w_1, w_2, w_2))$ must intersect with some boundary face of $DW(\mu I_n - B)$. Consequently, there are three vertices on the boundary face of $DW(\mu I_n - B)$ whose convex hull intersects with $DW(\operatorname{diag}(w_1, w_2, w_3))$. Now, for two triangular laminas each having vertices in $\mathbf{P}$ to have a nonempty intersection, there must be a nonempty intersection of a triangular lamina with an edge of another triangular lamina. By Lemma 3.3, there is a one-one correspondence between the triangles on the boundary faces of $DW(\mu I_n - B)$ and those on the boundary faces of $DW(B)$. Thus, condition (a) or (b) holds. □

One can also consider the boundary $\partial\mathcal{E}(A,B)$ of $\mathcal{E}(A,B)$. By Theorem 4.6 in section 4, if $A, B \in M_n$ are normal and each has at most two distinct eigenvalues,

then $\mathcal{E}(A,B)$ has an empty interior, i.e., $\partial\mathcal{E}(A,B) = \mathcal{E}(A,B)$. We will exclude these special cases. The following lemma is needed for further discussion.

LEMMA 3.5. [illegible] $S = \{w_1, w_2, w_3\}$ [illegible] $T = \{z_1, z_2\}$ [illegible] $\mathbb{C}$ [illegible]

$$\partial\mathcal{E}(S,T) = \mathcal{E}(\{w_1,w_2\},T) \cup \mathcal{E}(\{w_1,w_3\},T) \cup \mathcal{E}(\{w_2,w_3\},T).$$

[illegible]. Clearly, the result holds if S or T is a singleton. In the following, we may assume that $z_1 \neq z_2$. If $w_j = w_k$ for some $1 \leq j < k \leq 3$, then $\mathcal{E}(S,T) = \mathcal{E}(\{w_i, w_l\},T)$, where $l \notin \{j,k\}$, which has no interior point.

Suppose $w_1, w_2, w_3 \in \mathbb{C}$ are distinct. Let $X = \operatorname{diag}(w_1,w_2,w_3)$, $Y = \operatorname{diag}(z_1,z_2)$, and $X_{jk} = \operatorname{diag}(w_j, w_k)$ for $1 \leq j < k \leq 3$. By Theorem 2.2, $\mu \in \mathcal{E}(S,T)$ if and only if $DW(X) \cap DW(\mu I_2 - Y) \neq \emptyset$. Note that $DW(\mu I_2 - Y)$ is a line segment with vertices in $\mathbf{P}$ while $DW(X)$ is a triangular lamina with three edges, $DW(X_{12})$, $DW(X_{23})$, and $DW(X_{13})$. Thus, μ is a boundary point of $\mathcal{E}(S,T)$ if and only if the line segment $DW(\mu I_2 - Y)$ intersects the triangular lamina $DW(X)$ at its boundary, which is the union of line segments $DW(X_{12})$, $DW(X_{23})$, and $DW(X_{13})$. The result follows. □

By Lemma 3.5 and Theorem 3.2, we have the following theorem.

THEOREM 3.6. [illegible] $A, B \in M_n$ [illegible] 2 [illegible]

$$\partial\mathcal{E}(A,B) \subseteq \bigcup \{\mathcal{E}(S,T) : S \subseteq \sigma(A), T \subseteq \sigma(B) \text{ [illegible] } |S| = |T| = 2\}.$$

4. Essentially Hermitian matrices. Recall that a normal matrix is essentially Hermitian if all of its eigenvalues lie on a straight line. Let us warm up our discussion with the following results and examples on Hermitian matrices.

THEOREM 4.1. [illegible] $A, B \in M_n$ [illegible] $a_1 \geq a_2 \geq \cdots \geq a_n$ [illegible] $b_1 \geq b_2 \geq \cdots \geq b_n$ [illegible]

$$\mathcal{E}(A,B) = [a_n + b_n,\ a_1 + b_1] \setminus \bigcup_{j=1}^{n-1} (a_{j+1} + b_1,\ a_j + b_n) \cup (b_{j+1} + a_1,\ b_j + a_n),$$

[illegible] $(c,d) = \emptyset$, $c \geq d$

[illegible]. By Theorem 3.1(d), $\mu \notin \mathcal{E}(A,B)$ if and only if $\{a_1, a_2, \ldots, a_n\}$ can be separated from $\{\mu - b_1, \mu - b_2, \ldots, \mu - b_n\}$ by a circle. For $\mu \in \mathbb{R}$, this happens if and only if one of the following conditions is satisfied:

1. $\mu - b_1 > a_1 \Leftrightarrow \mu > a_1 + b_1$.
2. $\mu - b_n < a_n \Leftrightarrow \mu < a_n + b_n$.
3. For some $1 \leq j \leq n-1$, $a_{j+1} < \mu - b_1 \leq \mu - b_n < a_j \Leftrightarrow a_{j+1} + b_1 < \mu < a_j + b_n$.
4. For some $1 \leq j \leq n-1$, $\mu - b_j < a_n \leq a_1 < \mu - b_{j+1} \Leftrightarrow b_{j+1} + a_1 < \mu < b_j + a_n$.

Hence, the result follows. □

We have the following corollary.

COROLLARY 4.2. [illegible] $A, B \in M_n$ [illegible] 4.1 [illegible]

$$b_1 - b_n \geq \max_{1 \leq j \leq n-1} (a_j - a_{j+1}) \quad \text{[illegible]} \quad a_1 - a_n \geq \max_{1 \leq j \leq n-1} (b_j - b_{j+1}),$$

[illegible] $\mathcal{E}(A,B) = [a_n + b_n,\ a_1 + b_1]$

EXAMPLE 4.3. Suppose $n \geq 2$, $A, B \in M_n$ are Hermitian with eigenvalues $a_1 = 5$, $a_n = 2$, $b_1 = 4$, and $b_n = 1$. Then $\mathcal{E}(A, B) = [3, 9]$ is independent of the choices of a_i and b_j for $2 \leq i,\ j \leq n-1$.

EXAMPLE 4.4. Suppose $A, B \in M_3$ are Hermitian with eigenvalues $a_1 = 5$, $a_3 = 1$, $b_1 = 4$, and $b_3 = 2$. If $a_2 = 3$, then $\mathcal{E}(A, B) = [3, 9]$; if $a_2 \neq 3$, then $\mathcal{E}(A, B) \subsetneq [3, 9]$.

It is interesting to note that sometimes the set $\mathcal{E}(A, B)$ depends only on the extreme eigenvalues of A and B as shown in Example 4.3, but this is not always the case, as shown in Example 4.4.

In perturbation theory, if $A, B \in M_n$ are Hermitian such that $\|B\|$ is larger than the smallest singular value of A, then it may happen that $A + B$ is singular. However, if we know more about the eigenvalues of A and B, one can get a better perturbation bound.

EXAMPLE 4.5. Suppose $A, B \in M_n$ are Hermitian such that $\sigma(A) \subseteq \mathbb{R} \setminus (-r, s)$ for some $r, s \in (0, \infty)$ and $\sigma(B) \subseteq [-u, v]$ for some $u, v \in [0, \infty)$ such that $-r + v < 0$ and $-u + s > 0$. Then $A + B$ is invertible.

In [20, Theorem 2], Wielandt described a procedure to construct $\mathcal{E}(A, B)$ for a Hermitian matrix A and a skew-Hermitian matrix B with eigenvalues $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$. In particular, it was shown that the set $\mathcal{E}(A, B)$ is the intersection of all hyperbolic regions containing the set $\{a_j + b_k : 1 \leq j, k \leq n\}$. However, details of the proof were not given. In the following, we extend the result of Wielandt to any pair of essentially Hermitian matrices A and B. A detailed proof is given for the result.

To present the result and proof, we need some basic facts in the coordinate geometry of $\mathbb{R}^2$ (identified with $\mathbb{C}$). Suppose $w_1, w_2, z_1, z_2 \in \mathbb{C}$ such that $P = \operatorname{conv}\{w_r + z_s : r, s \in \{1, 2\}\}$ is a nondegenerate parallelogram. Then there is a unique rectangular hyperbola passing through the vertices of P. The hyperbola degenerates to a pair of perpendicular lines if and only if the four sides of P have equal length. Otherwise, each branch of the hyperbola will pass through a pair of vertices of P corresponding to a side of P with shorter length, i.e., the two sides of P of longer lengths lie in the closed region lying between the two branches of the hyperbola. For a nondegenerate rectangular hyperbola, the connected closed region with the hyperbola as boundary is the [illegible], and the two disconnected closed regions with the hyperbola as boundary form the [illegible]. Of course, the complement of a closed hyperbolic region is an open hyperbolic region, and vice versa. In case the hyperbola degenerates to a pair of perpendicular lines, the inner (and outer) hyperbolic region becomes the union of two unbounded triangular regions connected at their vertices.

Suppose A and B are two essentially Hermitian matrices. If the line through $\sigma(A)$ and the line through $\sigma(B)$ are parallel, then there are $\alpha, \beta \in \mathbb{C}$ and $\phi \in \mathbb{R}$ such that $H = e^{-i\phi}(A - \alpha I)$ and $K = e^{-i\phi}(B - \beta I)$ are Hermitian. Then

$$\mathcal{E}(A, B) = e^{i\phi}\mathcal{E}(H, K) + (\alpha + \beta)$$

and the result follows from Theorem 4.1. For the other cases, we have the following result.

THEOREM 4.6. [illegible] $A, B \in M_n$ [illegible] $\alpha, \beta \in \mathbb{C}$, $r_1 \geq r_2 \geq \cdots \geq r_n$, $s_1 \geq s_2 \geq \cdots \geq s_n$, $\phi, \theta \in \mathbb{R}$, [illegible] A, B [illegible] $a_j = \alpha + r_j e^{i\phi}$, $1 \leq j \leq n$, [illegible] $b_j = \beta + s_j e^{i\theta}$, $1 \leq j \leq n$, [illegible] $\Gamma = [r_n, r_1] \times [s_n, s_1]$ [illegible] $e^{i(\phi-\theta)} \notin \{1, -1\}$, [illegible]

(i) Let $S(a,b) = \{a_u + b_v : 1 \le u, v \le n\}$. For $1 \le j < n$, if $a_j \ne a_{j+1}$, then $S(a,b)$

$$H(a,j) = \{e^{i\phi}x + e^{i\theta}y + \alpha + \beta : (x,y) \in \mathbb{R}^2, (y-s_1)(y-s_n) \le (x-r_j)(x-r_{j+1})\};$$

If $b_j \ne b_{j+1}$, then $S(a,b)$

$$H(b,j) = \{e^{i\phi}x + e^{i\theta}y + \alpha + \beta : (x,y) \in \mathbb{R}^2, (y-s_j)(y-s_{j+1}) \ge (x-r_1)(x-r_n)\}.$$

(ii) If $\mathcal{E}(A,B)$ is nonempty, then $P = \operatorname{conv}\{a_r + b_s : r, s \in \{1,n\}\}$ by (i).

(iii) If $\mathcal{E}(A,B)$ is nonempty, then by (i) A and B are unitarily similar to $a_1 I_k \oplus A_1$ and $b_1 I_\ell \oplus B_1$, where $a_1 = \cdots = a_k \ne a_{k+1} = \cdots = a_n$ and $b_1 = \cdots = b_\ell \ne b_{\ell+1} = \cdots = b_n$. Then $\mathcal{E}(A,B)$

$$\begin{aligned}\mathcal{E}(A,B) &= P \cap H(a,k) \cap H(b,\ell)\\ &= \{e^{i\phi}x + e^{i\theta}y + \alpha + \beta : (x,y) \in \Gamma, (y-s_1)(y-s_n) = (x-r_1)(x-r_n)\}.\end{aligned}$$

Our proof depends on the following lemma.

LEMMA 4.7. Let $A, B \in M_n$ be as in Theorem 4.6 and $\mu \notin \mathcal{E}(A,B)$.

(a) $\mu - b_1, \mu - b_n$ lie on the same open side of the line through a_1, a_n.

(b) There exist $t_1, t_2 \in [0,1]$ and $j \in \{1, \ldots, n-1\}$ such that

$$\mu - (t_1 b_1 + (1-t_1)b_n) = t_2 a_j + (1-t_2)a_{j+1} \text{ and}$$

$$t_1|\mu - b_1|^2 + (1-t_1)|\mu - b_n|^2 < t_2|a_j|^2 + (1-t_2)|a_{j+1}|^2.$$

(c) There exist $t_1, t_2 \in [0,1]$ and $j \in \{1, \ldots, n-1\}$ such that

$$\mu - (t_1 b_j + (1-t_1)b_{j+1}) = t_2 a_1 + (1-t_2)a_n \text{ and}$$

$$t_1|\mu - b_j|^2 + (1-t_1)|\mu - b_{j+1}|^2 > t_2|a_1|^2 + (1-t_2)|a_n|^2.$$

Proof. Under the given assumption, $DW(A)$ and $DW(\mu I_n - B)$ will be vertical polygonal disks in $\mathbb{C} \times \mathbb{R}$ with vertices in $\{(z, |z|^2) : z \in \mathbb{C}\}$. The two disks have no intersection if and only if

(1) the projections of the two disks on $\mathbb{C}$ do not intersect, or

(2) the projections on $\mathbb{C}$ intersect, but one disk is above the other.

Case (1) is equivalent to (a), and (2) is equivalent to (b) or (c). □

Proof of Theorem 4.6. Suppose $\mu \notin \mathcal{E}(A,B)$. Consider the three cases in Lemma 4.7:

(a) The line segment joining a_1, a_n and the line segment joining $\mu - b_1, \mu - b_n$ have no intersection if and only if for all $0 \le t_1,\ t_2 \le 1$,

$$\begin{aligned} t_2a_1 + (1-t_2)a_n &\neq t_1(\mu - b_1) + (1-t_1)(\mu - b_n), \\ \Leftrightarrow \qquad \mu &\neq t_1b_1 + (1-t_1)b_n + t_2a_1 + (1-t_2)a_n, \\ \Leftrightarrow \qquad \mu &\notin P = \operatorname{conv}\{a_r + b_s : r, s \in \{1, n\}\}, \\ \Leftrightarrow \qquad \mu &\notin \{e^{i\phi}x + e^{i\theta}y + \alpha + \beta : (x, y) \in \Gamma\}. \end{aligned}$$

(b) Suppose for some $t_1, t_2 \in [0, 1]$ and $j \in \{1, \dots, n-1\}$ that

$$\mu - (t_1b_1 + (1-t_1)b_n) = t_2a_j + (1-t_2)a_{j+1} \tag{4.1}$$

and

$$t_1|\mu - b_1|^2 + (1-t_1)|\mu - b_n|^2 < t_2|a_j|^2 + (1-t_2)|a_{j+1}|^2. \tag{4.2}$$

Let $\mu - \alpha - \beta = e^{i\phi}u + e^{i\theta}v$ with $u, v \in \mathbb{R}$. From (4.1), $a_j = \alpha + e^{i\phi}r_j$, and $b_j = \beta + e^{i\theta}s_j$ for $1 \le j \le n$ with $e^{i(\phi-\theta)} \notin \{1, -1\}$, we have

$$u = t_2r_j + (1-t_2)r_{j+1} \quad \text{and} \quad v = t_1s_1 + (1-t_1)s_n$$

or, equivalently,

$$t_1 = \frac{s_n - v}{s_n - s_1} \quad \text{and} \quad t_2 = \frac{r_{j+1} - u}{r_{j+1} - r_j}.$$

We have

$$\begin{aligned} & t_2|a_j|^2 + (1-t_2)|a_{j+1}|^2 \\ &= t_2|\alpha + e^{i\phi}r_j|^2 + (1-t_2)|\alpha + e^{i\phi}r_{j+1}|^2 \\ &= t_2(|\alpha|^2 + (\overline{\alpha}e^{i\phi} + \alpha e^{-i\phi})r_j + r_j^2) + (1-t_2)(|\alpha|^2 + (\overline{\alpha}e^{i\phi} + \alpha e^{-i\phi})r_{j+1} + r_{j+1}^2) \\ &= |\alpha|^2 + (\overline{\alpha}e^{i\phi} + \alpha e^{-i\phi})u + (r_j + r_{j+1})u - r_jr_{j+1} \end{aligned}$$

as $t_2r_j + (1-t_2)r_{j+1} = u$ and $t_2r_j^2 + (1-t_2)r_{j+1}^2 = (r_j + r_{j+1})u - r_jr_{j+1}$. Then

$$\begin{aligned} & t_1|\mu - b_1|^2 + (1-t_1)|\mu - b_n|^2 \\ &= t_1|\alpha + e^{i\phi}u + e^{i\theta}(v - s_1)|^2 + (1-t_1)|\alpha + e^{i\phi}u + e^{i\theta}(v - s_n)|^2 \\ &= t_1\big[|\alpha|^2 + (\overline{\alpha}e^{i\phi} + \alpha e^{-i\phi})u + (\overline{\alpha}e^{i\theta} + \alpha e^{-i\theta})(v - s_1) \\ &\quad + (e^{i(\theta-\phi)} + e^{-i(\theta-\phi)})u(v - s_1) + u^2 + (v - s_1)^2\big] \\ &\quad + (1-t_1)\big[|\alpha|^2 + (\overline{\alpha}e^{i\phi} + \alpha e^{-i\phi})u + (\overline{\alpha}e^{i\theta} + \alpha e^{-i\theta})(v - s_n) \end{aligned}$$

$$+(e^{i(\theta-\phi)}+e^{-i(\theta-\phi)})u(v-s_n)+u^2+(v-s_n)^2\big]$$

$$=|\alpha|^2+(\overline{\alpha}e^{i\phi}+\alpha e^{-i\phi})u+u^2-(v-s_1)(v-s_n)$$

as $t_1(v-s_1)+(1-t_1)(v-s_n)=0$ and $t_1(v-s_1)^2+(1-t_1)(v-s_n)^2=-(v-s_1)(v-s_n)$.

Putting these values into (4.2), we have

$$0<t_2|a_j|^2+(1-t_2)|a_{j+1}|^2-t_1|\mu-b_1|^2-(1-t_1)|\mu-b_n|^2$$

$$=(v-s_1)(v-s_n)-(u-r_j)(u-r_{j+1}).$$

For any $z=e^{i\phi}x+e^{i\theta}y+\alpha+\beta$ with $x,y\in\mathbb{R}$, define

$$f(z)=(y-s_1)(y-s_n)-(x-r_j)(x-r_{j+1}).$$

With $a_k+b_m=e^{i\phi}r_k+e^{i\theta}s_m+\alpha+\beta$, we have

$$f(a_k+b_m)=(s_m-s_1)(s_m-s_n)-(r_k-r_j)(r_k-r_{j+1})\le 0.$$

Thus, $H(a,j)=\{z:f(z)\le 0\}$ is a closed hyperbolic region satisfying (i).

Similarly, if condition (c) in Lemma 4.7 is satisfied, then we have a closed hyperbolic region $H(b,j)$ satisfying (i).

By Lemma 4.7 and (i), we see that $\mathcal{E}(A,B)$ is a subset of the intersection of P and the hyperbolic regions described in (i), and no points in the complement of the intersection belong to $\mathcal{E}(A,B)$. Thus, assertion (ii) of the theorem follows.

From the above discussion, we can see that the complement of $\mathcal{E}(A,B)$ is a union of open hyperbolic regions. So, if $z\in\mathbb{C}\setminus\mathcal{E}(A,B)$, then there exists a half line L containing z with $L\cap\mathcal{E}(A,B)=\emptyset$. Hence, every connected component of $\mathcal{E}(A,B)$ is simply connected.

Suppose the boundary of the parallelogram $P=\operatorname{conv}\{a_u+b_v:u,v\in\{1,n\}\}$ is graduated by the points a_r+b_j and a_j+b_r with $r\in\{1,n\}$ and $j\in\{1,\dots,n\}$. Then the intersection of the hyperbolas $H(a,j)$ (respectively, $H(b,j)$) with P will have endpoints a_r+b_s with $r\in\{j,j+1\}$ and $s\in\{1,n\}$ (respectively, $r\in\{1,n\}$ and $s\in\{j,j+1\}$).

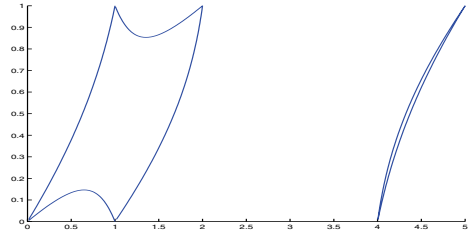
Combining the arguments in the last two paragraphs, we get condition (iii). □

4.8. The above result gives a simple procedure to determine the region $\mathcal{E}(A,B)$ for A and B satisfying the conditions in Theorem 4.6.

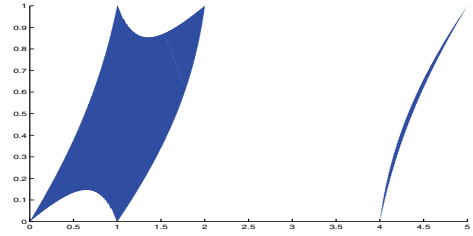
Sketch the hyperbolas corresponding to the intersection of P and the closed hyperbolic regions $H(a,j)$ and $H(b,j)$ for $1\le j<n$ (see section 6.2). Then $\mathcal{E}(A,B)$ consists of the simply connected regions in P determined by these curves.

4.9. Notice that all 2×2 normal matrices are essentially Hermitian. Then for any 2×2 nonscalar normal matrices A and B, $\mathcal{E}(A,B)$ is either a union of line segments or a pair of hyperbola by Theorems 4.1 and 4.6. In both cases, $\mathcal{E}(A,B)$ has an empty interior.

4.10. Consider $A=\operatorname{diag}(0,1,4)$ and $B=\operatorname{diag}(0,1+i)$. The following pictures depict the segments of hyperbolas corresponding to $H(a,1)$, $H(a,2)$, $H(b,1)$, and the set $\mathcal{E}(A,B)$.



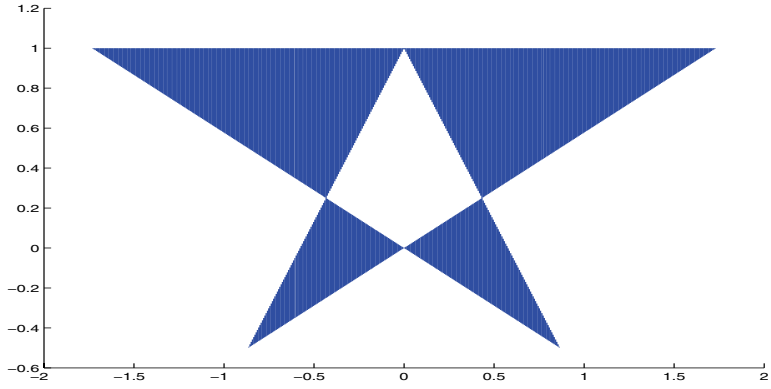
Example 4.10: $H(a,1),\cup H(a,2),\cup H(b,1)$



Example 4.10: $\mathcal{E}(A,B)$

Suppose $A, B \in M_n$ are normal matrices. The connected components of $\mathcal{E}(A,B)$ may not be simply connected in general as shown in the following example.

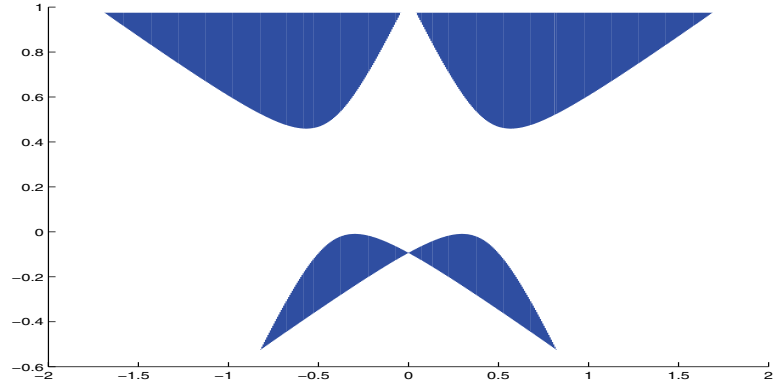
4.11. Let $\omega = e^{i2\pi/3}$. Using the method described in section 6, we can show that for $A = \operatorname{diag}(-i, -i\omega, -i\omega^2)$ and $B = \operatorname{diag}(-i\omega, -i\omega, -i\omega^2)$, $\mathcal{E}(A,B)$ is not simply connected.



Example 4.11: $\mathcal{E}(A,B)$

Although the conclusion of Theorem 4.6 does not hold for arbitrary normal matrices $A, B \in M_n$, one can see form Theorem 3.6 that the boundary of $\mathcal{E}(A,B)$ is a subset of the union of hyperbolas determined by eigenvalue pairs of A and eigenvalue pairs of B. We have the following example.

4.12. Here we let $\omega = e^{i2\pi/3}$, $A = \operatorname{diag}(-i, -i\omega, -i\omega^2)$, and $B = 0.95\operatorname{diag}(-i\omega, -i\omega, -i\omega^2)$. Then the boundary of $\mathcal{E}(A,B)$ are subsets of the union of hyperbolas.



Example 4.12: $\mathcal{E}(A,B)$

It is interesting to note that the matrices in Example 4.12 are obtained from those in Example 4.11 by shrinking B by a factor of 0.95, and hence the two pictures of $\mathcal{E}(A,B)$ have some resemblance even though part of the boundary changes from straight-line segments to curve segments. In general, it is not hard to show that $(A,B) \mapsto \mathcal{E}(A,B)$ is a continuous function, say, by using the usual topology on $M_n \times M_n$ and the Hausdorff metric for compact sets in $\mathbb{C}$.

5. Extensions and open problems. One may ask whether the results can be extended to the sum of k matrices from k different unitary similarity orbits for $k > 2$. For Hermitian matrices $A_1, \ldots, A_k$, there is a complete description of the eigenvalues of the matrices in $\mathcal{U}(A_1)+\cdots+\mathcal{U}(A_k)$; see [8]. For non-Hermitian matrices $A_1, \ldots, A_k \in M_n$, we can extend the idea in section 2 to determine the set of complex numbers μ, which is the eigenvalue of a matrix in $\mathcal{U}(A_1) + \cdots + \mathcal{U}(A_k)$. To this end, we need the concept of the modified Davis–Wielandt shell of $A \in M_n$ defined by

$$MDW(A) = \left\{ \left(x^*Ax, \sqrt{\|Ax\|^2 - |x^*Ax|^2} e^{it} \right) : x \in \mathbb{C}^n,\ x^*x = 1,\ t \in \mathbb{R} \right\} \subseteq \mathbb{C} \times \mathbb{C}.$$

Note that $(\mu_1, \mu_2) \in MDW(A)$ if and only if there is a unitary matrix U such that the first column of U^*AU equals $[\mu_1, \mu_2, 0, \ldots, 0]^t$.

THEOREM 5.1. [illegible] $A_1, \ldots, A_k \in M_n$ [illegible] $\mu \in \mathbb{C}$ [illegible]

(a) [illegible] $U_1, \ldots, U_k \in M_n$ [illegible] $\det(\sum_{j=1}^k U_j A_j U_j^* - \mu I_n) = 0$

(b) $(\mu, 0) \in MDW(A_1) + \cdots + MDW(A_k)$

(c) $[MDW(A_1) + \cdots + MDW(A_{k-1})] \cap MDW(\mu I_n - A_k) \neq \emptyset$

[illegible]. We may assume that $k \geq 3$. The implications $(c) \iff (b) \Rightarrow (a)$ are clear. Suppose (a) holds. Then there are unitary matrices $U_1, \ldots, U_k$ such that the first column of $\sum_{j=1}^k U_j^* A_j U_j$ equals $[\mu, 0, \ldots, 0]^t$. Let v_j be obtained from the first column of $U_j^* A_j U_j$ by removing its first entry μ_j. Then $\sum_{j=1}^k v_j = 0$. Relabel A_j so that $\|v_1\| \geq \cdots \geq \|v_k\|$. Then $\|v_1\| \leq \|v_2\| + \cdots + \|v_k\|$. Thus, there exist $t_1, \ldots, t_k \in \mathbb{R}$ such that $\sum_{j=1}^k \|v_j\| e^{it_j} = 0$. It follows that $(\mu_j, \|v_j\| e^{it_j}) \in MDW(A_j)$ for $j = 1, \ldots, k$ such that $(\mu, 0) = \sum_{j=1}^k (\mu_j, \|v_j\| e^{it_j})$. Thus, condition (b) holds. ∎

Besides the unitary similarity orbits, one may consider orbits of matrices under other group actions and consider the eigenvalues of the sum of matrices from different orbits.

For example, we can consider the usual similarity orbit of $A \in M_n$,

$$\mathcal{S}(A) = \{SAS^{-1} : S \in M_n \text{ is invertible}\};$$

the unitary equivalence orbit of $A \in M_n$,

$$\mathcal{V}(A) = \{UAV : U, V \in M_n \text{ are unitary}\};$$

and the unitary congruence orbit of $A \in M_n$,

$$\mathcal{U}^t(A) = \{UAU^t : U \in M_n \text{ is unitary}\}.$$

For example, if $A, B \in M_n$ are not scalar, then any $\mu \in \mathbb{C}$ can be an eigenvalue of $SAS^{-1} + B$. Can we prove this for complex orthogonal similarity?

One may also consider the eigenvalues of usual product, Lie product, and Jordan product of matrices from different orbits; e.g., see [10, 17]. Of course, one may address similar problems for matrices over reals or arbitrary fields or rings.

For example, our results in section 2 hold for real eigenvalues for real matrices $UAU^t + VBV^t$, where U, V are real orthogonal matrices.

6. Computer algorithms and programs. Using the result in section 2, we can use any positive semidefinite programming package to test whether $\mu \in \mathcal{E}(A, B)$ as follows. For every $(\xi, |\xi|^2) \in DW(\mu I - B)$, we check whether $(\xi, |\xi|^2) \in DW(A)$; equivalently, we check whether a real linear combination of the three Hermitian matrices

$$\mathrm{Re}(A - \xi I), \quad \mathrm{Im}(A - \xi I), \quad A^*A - |\xi|^2 I$$

is positive definite. (This can be done by any positive semidefinite programming package.) If there is no such combination, then $(\xi, |\xi|^2) \in DW(A)$.

Of course, the above test is inefficient and hard to implement. The situation will improve significantly for normal matrices. One can use any standard linear programming package to check whether the two convex polytopes $DW(A)$ and $DW(\mu I - B)$ have nonempty intersection.

The situation further improves if we use Theorem 3.4 and focus on $DW(X) \cap DW(\mu I_2 - Y)$ for normal matrices $X \in M_3$ and $Y \in M_2$. For convenience, we use $\mathcal{E}(X, Y)$ to denote the set of $\mu \in \mathbb{C}$ such that $DW(X) \cap DW(\mu I_2 - Y) \neq \emptyset$; even X and Y may not have the same size. Then the set $\mathcal{E}(A, B)$ is the union of $\mathcal{E}(X, Y)$, where $X = \mathrm{diag}\,(w_1, w_2, w_3) \in M_3$ and $Y = \mathrm{diag}\,(z_1, z_2) \in M_2$ as described in Theorem 3.4. Furthermore, if both A and B have only two distinct eigenvalues, respectively, say w_1, w_2 and z_1, z_2, then $\mathcal{E}(A, B) = \mathcal{E}(X, Y)$ with $X = \mathrm{diag}\,(w_1, w_2)$ and $Y = \mathrm{diag}\,(z_1, z_2)$.

In the following, we will focus on $\mathcal{E}(X, Y)$ so that either $(X, Y) \in M_2 \times M_2$ or $(X, Y) \in M_3 \times M_2$ with distinct eigenvalues. Also as $\mathcal{E}(X, Y)$ depends only on the eigenvalues of X and Y, we may assume that X and Y are diagonal in our discussion.

We describe an easy pointwise test for $x + iy \in \mathcal{E}(X, Y)$ in the following.

6.1. A pointwise test. The (2, 2) case. We begin with the simple case when $X = \mathrm{diag}\,(w_1, w_2), Y = \mathrm{diag}\,(z_1, z_2) \in M_2$, and determine whether a given point $x + iy \in \mathcal{E}(X, Y)$, for four given complex numbers $w_1 = a_1 + ib_1$, $w_2 = a_2 + ib_2$, $z_1 = c_1 + id_1$, and $z_2 = c_2 + id_2$ so that w_1, w_2 are distinct and z_1, z_2 are distinct.

Let $P_j = \left(a_j, b_j, a_j^2 + b_j^2\right)$ and $Q_j = \left(x - c_j, y - d_j, (x - c_j)^2 + (y - d_j)^2\right)$ for $j = 1, 2$. Then $x + iy \in \mathcal{E}$ if and only if

$$\overline{P_1P_2} \cap \overline{Q_1Q_2} \neq \emptyset. \tag{6.1}$$

Since all four points P_1, P_2, Q_1, Q_2 lie on the boundary of the convex set $\{(x, y, z) : x^2 + y^2 \leq z\} \subseteq \mathbb{R}^3$, (6.1) holds if and only if the four points lie on the same plane and P_1 and P_2 lie on the opposite closed half plane determined by the line through Q_1 and Q_2.

Let $\mathbf{u} = \overrightarrow{Q_1Q_2}$, $\mathbf{v} = \overrightarrow{Q_1P_2}$, and $\mathbf{r} = \mathbf{u} \times \mathbf{v} = (r_1, r_2, r_3)$. Define

$$\Delta_0 = \begin{vmatrix} c_1 - c_2 & a_1 + c_1 - x & a_2 - a_1 \\ d_1 - d_2 & b_1 + d_1 - y & b_2 - b_1 \\ (x-c_2)^2 + (y-d_2)^2 & a_1^2 + b_1^2 - (x-c_1)^2 - (y-d_1)^2 & a_2^2 + b_2^2 - a_1^2 - b_1^2 \\ -(x-c_1)^2 - (y-d_1)^2 & & \end{vmatrix},$$

$$\Delta_1 = \begin{vmatrix} c_1 - c_2 & a_1 + c_1 - x & r_1 \\ d_1 - d_2 & b_1 + d_1 - y & r_2 \\ (x-c_2)^2 + (y-d_2)^2 & a_1^2 + b_1^2 - (x-c_1)^2 - (y-d_1)^2 & r_3 \\ -(x-c_1)^2 - (y-d_1)^2 & & \end{vmatrix}.$$

Then P_1, P_2, Q_1, Q_2 all lie on the same plane if and only if $\Delta_0 = 0$. Suppose $\Delta_0 = 0$. Then P_1 and P_2 lie on the opposite closed half plane determined by the line through Q_1 and Q_2 if and only if $\Delta_1 \le 0$.

ASSERTION 6.1. [illegible] $X, Y \in M_2$ [illegible] $x + iy \in \mathcal{E}(X,Y)$ [illegible] $\Delta_0 = 0$ [illegible] $\Delta_1 \le 0$

The (3, 2) case. Next, we describe the test to determine whether a given point

$$x + iy \in \mathcal{E}\left(\operatorname{diag}(w_1, w_2, w_3), \operatorname{diag}(z_1, z_2)\right)$$

for any given complex numbers w_1, w_2, w_3, z_1, z_2 so that w_1, w_2, w_3 are distinct and z_1, z_2 are distinct. Let $w_j = a_j + ib_j$ for $j = 1, 2, 3$, and $z_k = c_k + id_k$ for $k = 1, 2$. Then $x + iy \in \mathcal{E}(X, Y)$ if and only if there exist $0 \le t_1 \le 1$, $0 \le t_2,\ t_3$ and $t_2 + t_3 \le 1$ such that

$$(1-t_1)\begin{pmatrix} x - c_1 \\ y - d_1 \\ (x-c_1)^2 + (y-d_1)^2 \end{pmatrix} + t_1 \begin{pmatrix} x - c_2 \\ y - d_2 \\ (x-c_2)^2 + (y-d_2)^2 \end{pmatrix}$$

$$= (1 - t_2 - t_3)\begin{pmatrix} a_1 \\ b_1 \\ a_1^2 + b_1^2 \end{pmatrix} + t_2 \begin{pmatrix} a_2 \\ b_2 \\ a_2^2 + b_2^2 \end{pmatrix} + t_3 \begin{pmatrix} a_3 \\ b_3 \\ a_3^2 + b_3^2 \end{pmatrix},$$

or, equivalently,

$$\begin{pmatrix} c_2 - c_1 & a_2 - a_1 & a_3 - a_1 \\ d_2 - d_1 & b_2 - b_1 & b_3 - b_1 \\ (x-c_1)^2 + (y-d_1)^2 & a_2^2 + b_2^2 - a_1^2 - b_1^2 & a_3^2 + b_3^2 - a_1^2 - b_1^2 \\ -(x-c_2)^2 - (y-d_2)^2 & & \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}$$

$$= \begin{pmatrix} x - c_1 - a_1 \\ y - d_1 - b_1 \\ (x-c_1)^2 + (y-d_1)^2 \\ -\left(a_1^2 + b_1^2\right) \end{pmatrix}.$$

Let

$$\Delta_0 = \begin{vmatrix} c_2-c_1 & a_2-a_1 & a_3-a_1 \\ d_2-d_1 & b_2-b_1 & b_3-b_1 \\ (x-c_1)^2+(y-d_1)^2 & a_2^2+b_2^2-a_1^2-b_1^2 & a_3^2+b_3^2-a_1^2-b_1^2 \\ -(x-c_2)^2-(y-d_2)^2 & & \end{vmatrix},$$

$$\Delta_1 = \begin{vmatrix} x-c_1-a_1 & a_2-a_1 & a_3-a_1 \\ y-d_1-b_1 & b_2-b_1 & b_3-b_1 \\ (x-c_1)^2+(y-d_1)^2 & a_2^2+b_2^2-a_1^2-b_1^2 & a_3^2+b_3^2-a_1^2-b_1^2 \\ -\left(a_1^2+b_1^2\right) & & \end{vmatrix},$$

$$\Delta_2 = \begin{vmatrix} c_2-c_1 & x-c_1-a_1 & a_3-a_1 \\ d_2-d_1 & y-d_1-b_1 & b_3-b_1 \\ (x-c_1)^2+(y-d_1)^2 & (x-c_1)^2+(y-d_1)^2 & a_3^2+b_3^2-a_1^2-b_1^2 \\ -(x-c_2)^2-(y-d_2)^2 & -\left(a_1^2+b_1^2\right) & \end{vmatrix},$$

$$\Delta_3 = \begin{vmatrix} c_2-c_1 & a_2-a_1 & x-c_1-a_1 \\ d_2-d_1 & b_2-b_1 & y-d_1-b_1 \\ (x-c_1)^2+(y-d_1)^2 & a_2^2+b_2^2-a_1^2-b_1^2 & (x-c_1)^2+(y-d_1)^2 \\ -(x-c_2)^2-(y-d_2)^2 & & -\left(a_1^2+b_1^2\right) \end{vmatrix}.$$

By the above discussion, we have the following.

ASSERTION 6.2. [illegible] $X \in M_3$ [illegible] $Y \in M_2$ [illegible] $\Delta_0 \neq 0$ [illegible] $x+iy \in \mathcal{E}(A,B)$ [illegible]

$$(\Delta_1, \Delta_2, \Delta_3, \Delta_0-\Delta_1, \Delta_0-\Delta_2-\Delta_3)/\Delta_0$$

[illegible]

Suppose $\Delta_0 = 0$. Let

$$P_j = \left(a_j, b_j, a_j^2+b_j^2\right) \text{ for } j = 1,2,3, \quad \text{and}$$

$$Q_k = \left(x-c_k, y-d_k, (x-c_k)^2+(y-d_k)^2\right) \text{ for } k = 1,2.$$

Then the line L through Q_1 and Q_2 is parallel to the plane Π determined by P_1, P_2, and P_3. Since all five points P_1, P_2, P_3, Q_1, Q_2 lie on the boundary of the convex set $\{(x,y,z) : x^2+y^2 \leq z\} \subseteq \mathbb{R}^3$, $x+iy \in \mathcal{E}(X,Y)$ if and only if L lies on Π and divides Π into two closed half planes with L as the common boundary such that each of these two closed half planes contains some P_i. Hence, L lies on Π if and only if $\Delta_0 = \Delta_1 = 0$. In such a case, let

$$\mathbf{u} = \overrightarrow{Q_1Q_2} = \left(c_1-c_2, d_1-d_2, (x-c_2)^2-(x-c_1)^2+(y-d_2)^2-(y-d_1)^2\right).$$

For $1 \leq j \leq 3$, let

$$\mathbf{v_j} = \overrightarrow{Q_1P_j} = \left(a_j-x+c_1, b_j-y+d_1, a_j^2+b_j^2-(x-c_1)^2-(y-d_1)^2\right).$$

If P_j and P_k lie on different half planes determined by L, then the cross products $\mathbf{u} \times \mathbf{v_j}$ and $\mathbf{u} \times \mathbf{v_k}$ are normals to Π, pointing in opposite directions. For $1 \leq j \leq 3$,

let $\mathbf{r_j} = \mathbf{u} \times \mathbf{v_j} = (r_{1j}, r_{2j}, r_{3j})$ and

$$\Delta'_j = \begin{vmatrix} a_2 - a_1 & a_3 - a_1 & r_{1j} \\ b_2 - b_1 & b_3 - b_1 & r_{2j} \\ a_2^2 + b_2^2 - a_1^2 - b_1^2 & a_3^2 + b_3^2 - a_1^2 - b_1^2 & r_{3j} \end{vmatrix}.$$

We can now describe the remaining case in the following.

ASSERTION 6.3. [illegible] $X \in M_3$ [illegible] $Y \in M_2$ [illegible] $\Delta_0 = 0$ [illegible] $x + iy \in \mathcal{E}(X, Y)$ [illegible] $\Delta_1 = 0$ [illegible] $\Delta'_j \le 0 \le \Delta'_k$ [illegible] $1 \le j, k \le 3$

Based on Assertions 6.1–6.3 with Theorem 3.4, we have written the MATLAB program PT.m (see http://www.math.wm.edu/~ckli/program/PT.m) to test whether a point $x + iy$ lies in $\mathcal{E}(A, B)$.

Also, if $A, B \in M_n$ are normal matrices, then $\mathcal{E}(A, B)$ is a subset of the set

$$\operatorname{conv}(\sigma(A) + \sigma(B)) = \operatorname{conv}\{a + b : a \in \sigma(A), b \in \sigma(B)\}.$$

One can then consider a grid in $\operatorname{conv}(\sigma(A)+\sigma(B))$ and apply the pointwise test to the grid points to plot $\mathcal{E}(A, B)$. The MATLAB program PPT.m (see http://www.math.wm.edu/~ckli/program/PPT.m) is written based on this idea. An example of $\mathcal{E}(A, B)$ generated by the program will be given in section 6.4.

6.2. Parametrization of $\mathcal{E}(A, B)$ for normal matrices. In this subsection, we give a parametrization of $\mathcal{E}(A, B)$. We start with the $(3, 2)$ case.

The (3, 2) case. Consider the case when $X = \operatorname{diag}(w_1, w_2, w_3) \in M_3$ and $Y = \operatorname{diag}(z_1, z_2) \in M_2$. Write $w_j = a_j + ib_j$ for $j = 1, 2, 3$ and $z_k = c_k + id_k$ for $k = 1, 2$. Let $P_j = \left(a_j, b_j, a_j^2 + b_j^2\right)$ for $j = 1, 2, 3$ and $Q_k = (x - c_k, y - d_k, (x - c_k)^2 + (y - d_k)^2)$ for $k = 1, 2$. As $\mu \in \mathcal{E}(X, Y)$ if and only if $\mu - w_1 - z_1 \in \mathcal{E}(X - w_1 I_3, Y - z_1 I_2)$, we may assume that $w_1 = z_1 = 0$, i.e., $a_1 = b_1 = c_1 = d_1 = 0$.

Notice that $\mathcal{E}(X, Y)$ is the set of $x + iy \in \mathbb{C}$ such that $\Delta(P_1P_2P_3) \cap \overline{Q_1Q_2} \neq \emptyset$. This holds if and only if there exists $0 \le t \le 1$ such that $\overline{P_1P_4} \cap \overline{Q_1Q_2} \neq \emptyset$, where

(6.2)
$$P_4 = (a_4, b_4, r_4) = \left(ta_2 + (1-t)a_3,\ tb_2 + (1-t)b_3,\ t(a_2^2 + b_2^2) + (1-t)(a_3^2 + b_3^2)\right).$$

By the convexity of the function $(x, y) \mapsto x^2 + y^2$, we have $r_4 \ge a_4^2 + b_4^2$. Thus, there are $0 \le t_1, t_2 \le 1$ such that

$$(1 - t_1)\begin{pmatrix} x \\ y \\ x^2 + y^2 \end{pmatrix} + t_1 \begin{pmatrix} x - c_2 \\ y - d_2 \\ (x - c_2)^2 + (y - d_2)^2 \end{pmatrix} = (1 - t_2)\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + t_2 \begin{pmatrix} a_4 \\ b_4 \\ r_4 \end{pmatrix},$$

or, equivalently,

$$x = c_2 t_1 + a_4 t_2, \qquad y = d_2 t_1 + b_4 t_2, \tag{6.3}$$

and

$$t_1(x^2 + y^2 - (x - c_2)^2 - (y - d_2)^2) + r_4 t_2 = x^2 + y^2. \tag{6.4}$$

Substituting (6.3) into (6.4) we get

$$(c_2^2+d_2^2)t_1(t_1-1)-(a_4^2+b_4^2)t_2^2+r_4t_2=0$$

which is a hyperbolic equation of t_1 and t_2 on $[0,1]$. Then

$$t_1=\frac{1}{2}\pm\sqrt{\frac{1}{4}+\left(\frac{a_4^2+b_4^2}{c_2^2+d_2^2}\right)t_2^2-\left(\frac{r_4}{c_2^2+d_2^2}\right)t_2}, \tag{6.5}$$

As $r_4\geq a_4^2+b_4^2$, it is easy to check that t_1 is well defined and $0\leq t_1\leq 1$ whenever $t_2\in[0,1]$ if $r_4^2\leq(a_4^2+b_4^2)(c_2^2+d_2^2)$, and

$$t_2\in\left[0,\frac{r_4-\sqrt{r_4^2-(a_4^2+b_4^2)(c_2^2+d_2^2)}}{2(a_4^2+b_4^2)}\right]\cup\left[\frac{r_4+\sqrt{r_4^2-(a_4^2+b_4^2)(c_2^2+d_2^2)}}{2(a_4^2+b_4^2)},1\right]$$

if $r_4^2>(a_4^2+b_4^2)(c_2^2+d_2^2)$.

ASSERTION 6.4. [illegible] $X\in M_3$ [illegible] $Y\in M_2$ [illegible] $t\in[0,1]$ [illegible] t_1 [illegible] t_2 [illegible] (6.2) [illegible] (6.5) [illegible] $x+iy\in\mathcal{E}(X,Y)$ [illegible] (6.3) [illegible] t_1 [illegible] t_2

The (2, 2) case. Next, we consider the $(2,2)$ case. By an argument similar to the $(3,2)$ case with $(a_4,b_4,r_4)=(a_2,b_2,a_2^2+b_2^2)$, we have

$$x=c_2t_1+a_2t_2,\qquad y=d_2t_1+b_2t_2, \tag{6.6}$$

and

$$(c_2^2+d_2^2)t_1(t_1-1)-(a_2^2+b_2^2)t_2(t_2-1)=0,$$

which is a hyperbolic equation of t_1 and t_2 on $[0,1]$. Then

$$t_1=\frac{1}{2}\pm\sqrt{\frac{1}{4}+\left(\frac{a_2^2+b_2^2}{c_2^2+d_2^2}\right)t_2(t_2-1)} \tag{6.7}$$

lies in $[0,1]$ whenever $t_2\in[0,1]$ if $c_2^2+d_2^2\geq a_2^2+b_2^2$, or

$$t_2=\frac{1}{2}\pm\sqrt{\frac{1}{4}+\left(\frac{c_2^2+d_2^2}{a_2^2+b_2^2}\right)t_1(t_1-1)} \tag{6.8}$$

lies in $[0,1]$ whenever $t_1\in[0,1]$ if $c_2^2+d_2^2<a_2^2+b_2^2$.

ASSERTION 6.5. [illegible] $X=\operatorname{diag}(0,w_2)\in M_2$ [illegible] $Y=\operatorname{diag}(0,z_2)\in M_2$ [illegible] $x+iy\in\mathcal{E}(X,Y)$ [illegible] (6.6) [illegible] t_1 [illegible] t_2 [illegible] (6.7) [illegible] (6.8)

Based on Assertions 6.4 and 6.5 and Theorem 3.4, we have written the MATLAB program HPT.m (see http://www.math.wm.edu/~ckli/program/HPT.m) to generate $\mathcal{E}(X,Y)$. An example of $\mathcal{E}(A,B)$ generated by the program will be given in section 6.4.

Using Theorem 3.6 and Assertion 6.5, we have written the MATLAB program BD32.m (see http://www.math.wm.edu/~ckli/program/BD32.m) to generate $\partial\mathcal{E}(X,Y)$, the boundary of $\mathcal{E}(X,Y)$ for normal $X\in M_3$ and $Y\in M_2$.

6.3. A different algorithm. To use the parametric approach in the previous subsection, one has to determine t_1 and t_2 in $[0,1]$, and draw two curves in terms of t_1 and t_2. Here, we introduce a different algorithm to generate $\mathcal{E} = \mathcal{E}(X,Y)$ with $X = \operatorname{diag}(w_1, w_2, w_3)$ and $Y = \operatorname{diag}(z_1, z_2)$. To generate the points $x+iy \in \mathcal{E}(X,Y)$, we first determine the range for x. Then for each x in the range, we determine the range of y. Here we consider the $(3,2)$ case only.

Let $w_j = a_j + ib_j$ and $z_k = c_k + id_k$ for $j = 1,2,3$ and $k = 1,2$. Since $\mathcal{E}(\mu X, \mu Y) = \mu\mathcal{E}(X,Y)$ and $\mathcal{E}(X+\nu I, Y-\nu I) = \mathcal{E}(X,Y)$, we may assume that $d_1 = d_2$. Also by a suitable relabeling, we can always assume $c_1 < c_2$ and $b_1 \le b_2 \le b_3$. Evidently, if $x+iy \in \mathcal{E}(X,Y)$, then

$$\min\{a_1,a_2,a_3\} + c_1 \le x \le \max\{a_1,a_2,a_3\} + c_2.$$

Now we choose an x satisfying the above inequalities and determine y so that $x+iy$ lies in $\mathcal{E}(X,Y)$.

In fact, except for the case when $b_1 = b_2 = b_3$, we may further assume that $b_1 < b_2 \le b_3$. In the exceptional case, $X - ib_1 I_3$ and $Y - id_1 I_2$ are Hermitian matrices. Then the result follows from Theorem 4.1. In detail, we have the following.

ASSERTION 6.6. [illegible] $a_1 < a_2 < a_3$, $b_1 = b_2 = b_3$, $c_1 < c_2$, [illegible] $d_1 = d_2$ [illegible] $x+iy \in \mathcal{E}(X,Y)$ [illegible] $y = b_1 + d_1$ [illegible]

$$x \in [a_1+c_1, a_3+c_2] \backslash (a_1+c_2, a_2+c_1) \cup (a_2+c_2, a_3+c_1) \cup (a_3+c_1, a_1+c_2).$$

From now on, we suppose that $b_1 < b_2 \le b_3$. As $\mathcal{E}(X - \mu I_3, Y + \mu I_2) = \mathcal{E}(X,Y)$, we can also assume that $|w_1| = |w_2| \ne |w_3|$ if w_1, w_2, w_3 are collinear and $|w_1| = |w_2| = |w_3|$, otherwise.

Note that as $d_1 = d_2$ and $|w_1| = |w_2|$, the determinants Δ_i defined in section 6.1 become

$$\Delta_0 = \begin{vmatrix} c_2 - c_1 & a_2 - a_1 & a_3 - a_1 \\ 0 & b_2 - b_1 & b_3 - b_1 \\ (x-c_1)^2 - (x-c_2)^2 & 0 & a_3^2 + b_3^2 - a_1^2 - b_1^2 \end{vmatrix},$$

$$\Delta_1 = \begin{vmatrix} x - c_1 - a_1 & a_2 - a_1 & a_3 - a_1 \\ y - d_1 - b_1 & b_2 - b_1 & b_3 - b_1 \\ (x-c_1)^2 + (y-d_1)^2 - (a_1^2 + b_1^2) & 0 & a_3^2 + b_3^2 - a_1^2 - b_1^2 \end{vmatrix},$$

$$\Delta_2 = \begin{vmatrix} c_2 - c_1 & x - c_1 - a_1 & a_3 - a_1 \\ 0 & y - d_1 - b_1 & b_3 - b_1 \\ (x-c_1)^2 - (x-c_2)^2 & (x-c_1)^2 + (y-d_1)^2 - (a_1^2 + b_1^2) & a_3^2 + b_3^2 - a_1^2 - b_1^2 \end{vmatrix},$$

$$\Delta_3 = \begin{vmatrix} c_2 - c_1 & a_2 - a_1 & x - c_1 - a_1 \\ 0 & b_2 - b_1 & y - d_1 - b_1 \\ (x-c_1)^2 - (x-c_2)^2 & 0 & (x-c_1)^2 + (y-d_1)^2 - (a_1^2 + b_1^2) \end{vmatrix},$$

where the last three determinants can been expressed in the form

$$\Delta_i = \Delta_{i2}(y-d_1)^2 + \Delta_{i1}(y-d_1) + \Delta_{i0} \quad i = 1,2,3,$$

with

$$\Delta_{12} = \begin{vmatrix} a_2 - a_1 & a_3 - a_1 \\ b_2 - b_1 & b_3 - b_1 \end{vmatrix},$$

$$\Delta_{11} = -\begin{vmatrix} a_2 - a_1 & a_3 - a_1 \\ 0 & a_3^2 + b_3^2 - a_1^2 - b_1^2 \end{vmatrix},$$

$$\Delta_{10} = \begin{vmatrix} x - c_1 - a_1 & a_2 - a_1 & a_3 - a_1 \\ -b_1 & b_2 - b_1 & b_3 - b_1 \\ (x-c_1)^2 - (a_1^2 + b_1^2) & 0 & a_3^2 + b_3^2 - a_1^2 - b_1^2 \end{vmatrix},$$

$$\Delta_{22} = -\begin{vmatrix} c_2 - c_1 & a_3 - a_1 \\ 0 & b_3 - b_1 \end{vmatrix},$$

$$\Delta_{21} = \begin{vmatrix} c_2 - c_1 & a_3 - a_1 \\ (x-c_1)^2 - (x-c_2)^2 & a_3^2 + b_3^2 - a_1^2 - b_1^2 \end{vmatrix},$$

$$\Delta_{20} = \begin{vmatrix} c_2 - c_1 & x - c_1 - a_1 & a_3 - a_1 \\ 0 & -b_1 & b_3 - b_1 \\ (x-c_1)^2 - (x-c_2)^2 & (x-c_1)^2 - (a_1^2 + b_1^2) & a_3^2 + b_3^2 - a_1^2 - b_1^2 \end{vmatrix},$$

$$\Delta_{32} = \begin{vmatrix} c_2 - c_1 & a_2 - a_1 \\ 0 & b_2 - b_1 \end{vmatrix},$$

$$\Delta_{31} = -\begin{vmatrix} c_2 - c_1 & a_2 - a_1 \\ (x-c_1)^2 - (x-c_2)^2 & 0 \end{vmatrix},$$

$$\Delta_{30} = \begin{vmatrix} c_2 - c_1 & a_2 - a_1 & x - c_1 - a_1 \\ 0 & b_2 - b_1 & -b_1 \\ (x-c_1)^2 - (x-c_2)^2 & 0 & (x-c_1)^2 - (a_1^2 + b_1^2) \end{vmatrix}.$$

Note that

$$\Delta_0 = (c_2 - c_1)\begin{vmatrix} b_2 - b_1 & b_3 - b_1 \\ 0 & a_3^2 + b_3^2 - a_1^2 - b_1^2 \end{vmatrix} + ((x-c_1)^2 - (x-c_2)^2)\begin{vmatrix} a_2 - a_1 & a_3 - a_1 \\ b_2 - b_1 & b_3 - b_1 \end{vmatrix}.$$

Therefore, $\Delta_0 = 0$ if and only if

$$w_1, w_2, w_3 \text{ are not collinear and } x = (c_1 + c_2)/2. \tag{6.9}$$

Suppose (6.9) holds. Then $\Delta_0 = 0$ and by Assertion 6.3, $x + iy \in \mathcal{E}(X, Y)$ only if $\Delta_1 = 0$, in which the equality holds when

$$y = d_1 \pm \sqrt{(x - c_1)^2 - (a_1^2 + b_1^2)}.$$

Now we can check whether the point $x + iy$ in $\mathcal{E}(X, Y)$ holds by considering the values of Δ_i' defined in Assertion 6.3.

Exclude the above case. Then $\Delta_0 \neq 0$. By Assertion 6.2, $x + iy \in \mathcal{E}(X, Y)$ if and only if

$$\Delta_1/\Delta_0 \geq 0, \quad \Delta_2/\Delta_0 \geq 0, \quad \Delta_3/\Delta_0 \geq 0,$$
$$(\Delta_0 - \Delta_1)/\Delta_0 \geq 0, \quad \text{and} \quad (\Delta_0 - \Delta_2 - \Delta_3)/\Delta_0 \geq 0.$$

In the following, we determine the possible range of y that satisfies the above inequalities.

Suppose $\alpha_1 \leq \beta_1, \ldots, \alpha_5 \leq \beta_5$ are the real solutions, if they exist, of the following quadratic equations:

$$\Delta_1 = \Delta_{12}(y - d_1)^2 + \Delta_{11}(y - d_1) + \Delta_{10} = 0, \tag{6.10}$$

TABLE 1

(6.11)	(6.12)	$\Delta_0 > 0$	$\Delta_0 < 0$
Y	Y	$[\alpha_2, \beta_2] \backslash (\alpha_3, \beta_3)$	$[\alpha_3, \beta_3] \backslash (\alpha_2, \beta_2)$
Y	N	$[\alpha_2, \beta_2]$	no solution
N	Y	no solution	$[\alpha_3, \beta_3]$
N	N	no solution	no solution

TABLE 2

	$b_2 \neq b_3 \quad (\Delta_{22} + \Delta_{32} \neq 0)$		$b_2 = b_3 \quad (\Delta_{22} + \Delta_{32} = 0)$	
(6.14)	$\Delta_0 > 0$	$\Delta_0 < 0$	$(\Delta_{21} + \Delta_{31})/\Delta_0 > 0$	$(\Delta_{21} + \Delta_{31})/\Delta_0 < 0$
Y	$(-\infty, \alpha_5] \cup [\beta_5, \infty)$	$[\alpha_5, \beta_5]$	$(-\infty, \alpha_5]$	$[\alpha_5, \infty)$
N	$(-\infty, \infty)$	no solution	/	/

$$\Delta_2 = \Delta_{22}(y - d_1)^2 + \Delta_{21}(y - d_1) + \Delta_{20} = 0, \tag{6.11}$$

$$\Delta_3 = \Delta_{32}(y - d_1)^2 + \Delta_{31}(y - d_1) + \Delta_{30} = 0, \tag{6.12}$$

$$\Delta_0 - \Delta_1 = -\Delta_{12}(y - d_1)^2 - \Delta_{11}(y - d_1) - \Delta_{10} + \Delta_0 = 0, \tag{6.13}$$

$$\begin{aligned}\Delta_0 - \Delta_2 - \Delta_3 = -(\Delta_{22} + \Delta_{32})(y - d_1)^2 - (\Delta_{21} + \Delta_{31})(y - d_1) \\ -(\Delta_{20} + \Delta_{30}) + \Delta_0 = 0.\end{aligned} \tag{6.14}$$

Also, we continue to use α_i to denote the corresponding real solution if the quadratic equation is linear.

As $b_1 < b_2 \leq b_3$,

$$\Delta_{22} = -(c_2 - c_1)(b_3 - b_1) < 0 \quad \text{and} \quad \Delta_{32} = (c_2 - c_1)(b_2 - b_1) > 0.$$

Thus, the inequalities $\Delta_2/\Delta_0 \geq 0$ and $\Delta_3/\Delta_0 \geq 0$ are satisfied if and only if y lies in the interval specified in Table 1 where "Y" denotes the corresponding equation having real solution(s) and "N" otherwise.

Now we turn to (6.14). Note that

$$\Delta_{22} + \Delta_{32} = \begin{vmatrix} c_2 - c_1 & a_2 - a_3 \\ 0 & b_2 - b_3 \end{vmatrix} \leq 0.$$

Therefore the equation is linear, equivalently $\Delta_{22} + \Delta_{32} = 0$, if and only if $b_2 = b_3$, which can hold only if w_1, w_2, w_3 are not collinear. In this case, $a_3^2 + b_3^2 - a_1^2 - b_1^2 = |w_3|^2 - |w_1|^2 = 0$ and so

$$\Delta_{21} + \Delta_{31} = \begin{vmatrix} c_2 - c_1 & a_3 - a_2 \\ (x - c_1)^2 - (x - c_2)^2 & a_3^2 + b_3^2 - a_1^2 - b_1^2 \end{vmatrix} \neq 0.$$

Therefore the inequality $(\Delta_0 - \Delta_2 - \Delta_3)/\Delta_0 \geq 0$ is satisfied if and only if y lies in the intervals specified in Table 2.

TABLE 3

		Noncollinear ($\Delta_{12} \neq 0$)		Collinear ($\Delta_{12} = 0$)		
(6.10)	(6.13)	$\Delta_{12}/\Delta_0 > 0$	$\Delta_{12}/\Delta_0 < 0$	$\Delta_{11}/\Delta_0 > 0$	$\Delta_{11}/\Delta_0 = 0$	$\Delta_{11}/\Delta_0 < 0$
Y	Y	$[\alpha_4, \beta_4] \backslash (\alpha_1, \beta_1)$	$[\alpha_1, \beta_1] \backslash (\alpha_4, \beta_4)$	$[\alpha_1, \alpha_4]$	$(-\infty, \infty)$	$[\alpha_4, \alpha_1]$
Y	N	no solution	$[\alpha_1, \beta_1]$	/	/	/
N	Y	$[\alpha_4, \beta_4]$	no solution	/	/	/
N	N	no solution	no solution	/	/	/

Finally, we consider (6.10) and (6.13). Clearly, the equations are linear, i.e., $\Delta_{12} = 0$, if and only if w_1, w_2, w_3 are collinear. In addition, the equations are constant functions, i.e., $\Delta_{12} = 0$ and $\Delta_{11} = 0$, if and only if $a_1 = a_2 = a_3$. In the case of being a constant function,

$$\Delta_0 = (c_2 - c_1)(b_2 - b_1)(a_3^2 + b_3^2 - a_1^2 - b_1^2) \quad \text{and} \quad \Delta_1 = (x - c_1 - a_1)(b_2 - b_1)(a_3^2 + b_3^2 - a_1^2 - b_1^2).$$

Thus, the inequalities $\Delta_1/\Delta_0 \geq 0$ and $(\Delta_0 - \Delta_1)/\Delta_0 \geq 0$ are satisfied if and only if $c_1 \leq x - a_1 \leq c_2$, which always holds by our assumption on x.

Combining with the quadratic and linear cases, the inequalities $\Delta_1/\Delta_0 \geq 0$ and $(\Delta_0 - \Delta_1)/\Delta_0 \geq 0$ are satisfied if and only if y lies in the intervals specified in Table 3.

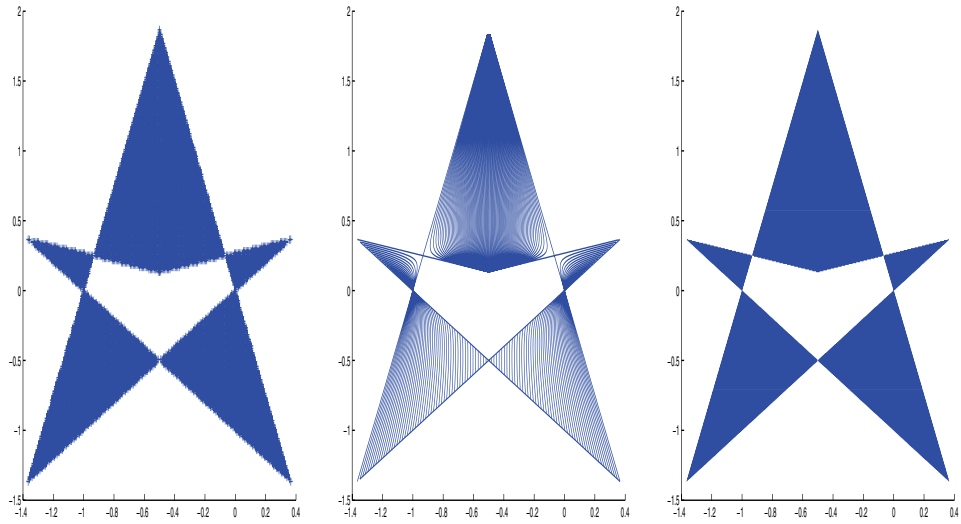
ASSERTION 6.7. [illegible] $b_1 < b_2 \leq b_3$, $c_1 < c_2$, $d_1 = d_2$ [illegible] (i) $|w_1| = |w_2| \neq |w_3|$, w_1, w_2, w_3 [illegible] (ii) $|w_1| = |w_2| = |w_3|$ [illegible] (6.9) [illegible] $x \in [a_{\min} + c_1, a_{\max} + c_2]$ [illegible] $a_{\min} = \min\{a_1, a_2, a_3\}$ [illegible] $a_{\max} = \max\{a_1, a_2, a_3\}$, $x + iy \in \mathcal{E}(X, Y)$ [illegible] y [illegible] 1, 2 [illegible] 3

Based on Assertions 6.6–6.7, we have written another MATLAB program IPT.m (see http://www.math.wm.edu/~ckli/program/IPT.m) to generate $\mathcal{E}(A, B)$ for normal matrices A and B. An example of $\mathcal{E}(A, B)$ generated by the program will be given in section 6.4.

6.4. An example of $\mathcal{E}(A, B)$ generated by the three approaches.

[illegible] 6.8. Let $A = \text{diag}\,(i, i\omega, i\omega^2)$ and $B = \text{diag}\,(\omega, \omega^2)$ with $\omega = e^{i2\pi/3}$. The region of $\mathcal{E}(A, B)$ is plotted using MATLAB programs based on the three different algorithms in sections 6.1–6.3.

In the above example, we see that the first program took the longest computer time and a lot of memory to determine and store $\mathcal{E}(A, B)$. The second program took less computer time and less memory, but it is not effective in approximating the straight line boundary of $\mathcal{E}(A, B)$ (using hyperbolas). Finally, the third program used the least amount of computer time and memory to produce and store $\mathcal{E}(A, B)$.



$\mathcal{E}(A,B)$ plotted by PPT.m $\mathcal{E}(A,B)$ plotted by HPT.m $\mathcal{E}(A,B)$ plotted by IPT.m

REFERENCES

[1] M. Adam and M. J. Tsatsomeros, *An eigenvalue inequality and spectrum localization for complex matrices*, Electron. J. Linear Algebra, 15 (2006), pp. 239–250.

[2] N. Ali Amir-Moéz, *Extreme Properties of Linear Transformations*, Polygonal Publishing, Washington, NJ, 1990.

[3] I. Bendixson, *Sur les racine d'une équation fondamentale*, Acta Math. 25 (1902), pp. 359–365.

[4] R. Bhatia, *Perturbation Bounds for Matrix Eigenvalues*, in Pitman Research Notes in Mathematics 162, Longman Scientific and Technical, New York, 1987.

[5] C. Davis, *The shell of a Hilbert-space operator*, Acta Sci. Math. (Szeged), 29 (1968), pp. 69–86.

[6] C. Davis, *The shell of a Hilbert-space operator*, II, Acta Sci. Math. (Szeged), 31 (1970), pp. 301–318.

[7] I. M. Gelfand and M. Naimark, *The relations between the unitary representations of the complex unimodular group and its unitary subgroups*, Izv. Akad. Nauk SSSR Ser. Mat., 14 (1950), pp. 239–260.

[8] W. Fulton, *Eigenvalues, invariant factors, highest weights, and Schubert calculus*, Bull. Amer. Math. Soc., 37 (2000), pp. 209–249.

[9] W. Fulton, *Eigenvalues of sums of Hermitian matrices* (after A. Klyachko), Seminaire Bourbaki, Vol. 1997/98. Astérisque No. 252 (1998), pp. 255–269.

[10] S. Furtado, L. Iglésias, and F. C. Silva, *Products of matrices with prescribed spectra and rank*, Linear Algebra Appl., 340 (2002), pp. 137–147.

[11] A. Horn, *Eigenvalues of sums of Hermitian matrices*, Pacific J. Math., 12 (1962), pp. 225–241.

[12] A. A. Klyachko, *Stable bundles, representation theory and Hermitian operators*, Selecta Math. (N.S.), 4 (1998), pp. 419–445.

[13] A. Knutson and T. Tao, *The honeycomb model of* $\mathrm{GL}_n(c)$ *tensor products,* I, *Proof of the saturation conjecture*, J. Amer. Math. Soc., 12 (1999), pp. 1055–1090.

[14] S. R. Lay, *Convex Sets and their Applications*, Pure Appl. Math., John Wiley & Sons, Inc., New York, 1982.

[15] C. K. Li, Y. T. Poon, and N. K. Sze, *Davis-Wielandt shells of operators*, Operators and Matrices, to appear.

[16] W. V. Parker, *Characteristic roots of a set of matrices*, Amer. Math. Monthly, 60 (1953), pp. 247–250.

[17] E. A. Martins and F. C. Silva, *On the eigenvalues of Jordan products*, Linear Algebra Appl., 359 (2003), pp. 249–262.

[18] G. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, in Computer Science and Scientific Computing, Academic Press, Boston, 1990.

[19] R. C. Thompson and L. Freede, *On the eigenvalues of sums of Hermitian matrices*, Linear Algebra Appl., 4 (1971), pp. 369–376.

[20] H. Wielandt, *On eigenvalues of sums of normal matrices*, Pacific J. Math., 5 (1955), pp. 633–638.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 582–608

COMBINATION PRECONDITIONING AND THE BRAMBLE–PASCIAK$^+$ PRECONDITIONER*

MARTIN STOLL† AND ANDY WATHEN†

Abstract. It is widely appreciated that the iterative solution of linear systems of equations with large sparse matrices is much easier when the matrix is symmetric. It is equally advantageous to employ symmetric iterative methods when a nonsymmetric matrix is self-adjoint in a nonstandard inner product. Here, general conditions for such self-adjointness are considered. A number of known examples for saddle point systems are surveyed and combined to make new combination preconditioners which are self-adjoint in different inner products. In particular, a new method related to the Bramble–Pasciak CG method is introduced and it is shown that a combination of the two outperforms the widely used classical method on a number of examples. Furthermore, we combine Bramble and Pasciak's method with a recently introduced method by Schöberl and Zulehner. The result gives a new preconditioner and inner product that can outperform the original methods of Bramble–Pasciak and Schöberl–Zulehner.

Key words. linear systems, Krylov subspaces, nonstandard inner products

AMS subject classifications. Primary, 65F10, 65N22, 65F50; Secondary, 76D07

DOI. 10.1137/070688961

1. Introduction.

In 1988, Bramble and Pasciak [6] introduced a block triangular preconditioner for the discrete Stokes problem (matrix) which had the almost magical effect of turning the original indefinite symmetric matrix problem into a nonsymmetric matrix which is both self-adjoint and, in certain practical circumstances, positive definite in a nonstandard inner product; thus the conjugate gradient method could be used in the nonstandard inner product.

Precisely, the symmetric saddle point problem

$$\underbrace{\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}}_{\mathcal{A}} x = b \tag{1.1}$$

with symmetric positive definite $A \in \mathbb{R}^{n\times n}$, symmetric positive semidefinite $C \in \mathbb{R}^{m\times m}$, $m < n$, and $B \in \mathbb{R}^{m\times n}$ of row full rank m, if preconditioned on the left by

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ B & -I \end{bmatrix} \text{ with } \mathcal{P}^{-1} = \begin{bmatrix} A_0^{-1} & 0 \\ BA_0^{-1} & -I \end{bmatrix}, \tag{1.2}$$

results in the nonsymmetric matrix

$$\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A} = \begin{bmatrix} A_0^{-1}A & A_0^{-1}B^T \\ BA_0^{-1}A - B & BA_0^{-1}B^T + C \end{bmatrix}, \tag{1.3}$$

*Received by the editors April 23, 2007; accepted for publication (in revised form) by M. Benzi February 7, 2008; published electronically June 6, 2008.
http://www.siam.org/journals/simax/30-2/68896.html
†Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, UK (martin.stoll@comlab.ox.ac.uk, andy.wathen@comlab.ox.ac.uk).

which turns out to be self-adjoint (many would say "symmetric") in the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ defined by $\langle w, y \rangle_{\mathcal{H}} := w^T \mathcal{H} y$, where

$$\mathcal{H} = \begin{bmatrix} A - A_0 & 0 \\ 0 & I \end{bmatrix}.$$

Moreover, $\langle w, \widehat{\mathcal{A}} w \rangle_{\mathcal{H}} > 0$ for all $w \neq 0$ so that $\widehat{\mathcal{A}}$ is also positive definite. For these results to hold, the matrix block A_0 has to be symmetric and positive definite and must be scaled in order that $A - A_0$ is also positive definite so that $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ not only defines a symmetric bilinear form but also satisfies the positivity requirement $\langle x, x \rangle_{\mathcal{H}} > 0$ for $x \neq 0$ which ensures that it is an inner product.

The outcome is that the conjugate gradient (CG) method [26] based on this inner product can be applied and the system can be solved efficiently; see, for example, Algorithm 2.1 in [11] for the CG method in an arbitrary inner product. This Bramble–Pasciak CG method is a very powerful and widely used tool to solve saddle point systems. Further analysis and applications can be found in [47, 9, 29, 28, 7, 1, 48, 22, 2, 3, 37].

The Bramble–Pasciak CG method is not the only solver of this kind where a matrix is preconditioned and then a nonstandard inner product can be found such that efficient Krylov subspace solvers like CG can be applied. More examples are given in [15, 42, 5, 34, 28, 31, 40, 8] and will be explained and used later in this paper.

We comment that alternative approaches for the Stokes and other saddle point problems use symmetric preconditioners and iterative methods such as MINRES and ITFQMR (see [11, Chapter 6], [4, 17]). We show some comparisons with such methods; see also [9]. We explore some abstract but elementary algebraic structures which enable some broadening of the set of available preconditioners for which saddle point problems may be treated by symmetric iterative methods in nonstandard inner-products. The outcome is some new preconditioning techniques which might be useful in practice. In particular, we derive and test numerically a new method which requires exactly the same work per iteration as the widely used Bramble–Pasciak method but which converges in fewer iterations for several examples we have computed. We furthermore introduce a method that combines the Bramble–Pasciak method and a recently introduced method by Schöberl and Zulehner with promising numerical results. Our algebraic results apply generally but we have not considered anything other than saddle point examples.

2. Background. For linear systems with large dimensions, it is well known that iterative methods are most often the only feasible solution approaches; direct methods for dense and sparse or structured matrices work well provided the bandwidth/skyline is not too large, but if fill-in in the computed triangular factors is too great such methods are usually infeasible beyond a certain dimension. Amongst the available iterative methods, multigrid approaches are extremely attractive for certain classes of problems (see, for example, [11]), and in more generality, Krylov subspace methods can be excellent solvers provided suitably fast convergence can be achieved; preconditioning is almost always required to achieve this.

For symmetric matrix systems, the CG method [26] for positive definite systems and minimum residual (MINRES) or SYMMLQ methods [35] for indefinite systems are based on short term recurrences and are the Krylov subspace methods of choice. The CG method is especially popular because of its efficiency. By contrast, for nonsymmetric matrix systems there is a large number of methods (GMRES, BICGSTAB, QMR; see [39, 45, 19]) and various hybrid methods (GMRESR, BICGSTAB(ℓ); see [46, 44])

reflecting that convergence is not as well understood than in the symmetric case and the best method for any particular problem is usually not clear; see [43] for a good overview and further references. This situation is rather unsatisfactory. Certainly if it were computationally possible to somehow convert a nonsymmetric system to an equivalent one with a symmetric matrix so that CG or MINRES or some other symmetric method could be employed then this could be very attractive. Even in the case of a symmetric and indefinite matrix system, conversion to a symmetric and positive definite matrix system is attractive since it enables use of CG (the above mentioned case of the Bramble–Pasciak method is an example). Use of the normal equations approach—replacing $Ax = b$ with $A^TAx = A^Tb$—is usually not so attractive because for other than a matrix which is very well conditioned this usually leads to much poorer convergence of iterative methods.

One can broaden the possibilities by allowing nonstandard inner-products—the important/desirable matrices are then the matrices which are self-adjoint with respect to such an inner product. The self-adjointness in a nonstandard inner-product enables the use of CG for positive definite systems and MINRES for self-adjoint but indefinite systems in that inner product. A method like the ideal transpose-free quasi minimal residuals (ITFQMR) [17] can be used if a matrix is only self-adjoint in a symmetric bilinear form (cf. section 7.4). For related considerations, in particular in connection with Krylov subspace methods, see [43, section 13].

Here, we use the term symmetric when referring to matrices which are self-adjoint in the usual Euclidean inner product defined by $\langle w, y\rangle := w^Ty = \sum w_iy_i$, i.e., those matrices whose entries satisfy $a_{i,j} = a_{j,i}$. Correspondingly, by $A^T(= B)$ we mean the matrix with entries $b_{i,j} = a_{j,i}$, i.e., the adjoint matrix in the usual Euclidean inner product.

3. Basic properties. First, we review the basic mathematics. We consider here only real Euclidean vector spaces; we see no reason that our theory should not apply in the complex case or indeed for other vector spaces, but we have not done so.

We say that

$$\langle\cdot,\cdot\rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \tag{3.1}$$

is a symmetric bilinear form if

- $\langle w, y\rangle = \langle y, w\rangle$ for all $w, y \in \mathbb{R}^n$
- $\langle \alpha w + y, z\rangle = \alpha\langle w, z\rangle + \langle y, z\rangle$ for all $w, y, z \in \mathbb{R}^n$ and all $\alpha \in \mathbb{R}$.

With the addition of a nondegeneracy condition, Gohberg, Lancaster, and Rodman (cf. [23]) use the term "indefinite inner product"; general properties of such forms can also be found here.

If, additionally, the positivity conditions

$$\langle w, w\rangle > 0 \text{ for } w \neq 0 \quad \text{with} \quad \langle w, w\rangle = 0 \text{ if and only if } w = 0$$

are satisfied, then (3.1) defines an inner product on $\mathbb{R}^n$.

For any real symmetric matrix, $\mathcal{H}$, $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ defined by

$$\langle w, y\rangle_{\mathcal{H}} := w^T\mathcal{H}y \tag{3.2}$$

is easily seen to be a symmetric bilinear form which is an inner product if and only if $\mathcal{H}$ is positive definite.

A matrix $\mathcal{A} \in \mathbb{R}^{n\times n}$ is self-adjoint in $\langle\cdot,\cdot\rangle$ if and only if

$$\langle \mathcal{A}w, y\rangle = \langle w, \mathcal{A}y\rangle \qquad \text{for all } w, y.$$

Self-adjointness of the matrix $\mathcal{A}$ in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ thus means that

$$w^T\mathcal{A}^T\mathcal{H}y = \langle \mathcal{A}w, y\rangle_{\mathcal{H}} = \langle w, \mathcal{A}y\rangle_{\mathcal{H}} = w^T\mathcal{H}\mathcal{A}y$$

for all w, y so that

$$\mathcal{A}^T\mathcal{H} = \mathcal{H}\mathcal{A} \tag{3.3}$$

is the basic relation for self-adjointness of $\mathcal{A}$ in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$.

Furthermore, we want to describe basic properties of bilinear forms and nonstandard inner products. This can also be viewed in terms of real symmetric matrices since (3.3) states that $\mathcal{A}^T\mathcal{H}$ is a real symmetric matrix. Here, we prefer the language of inner products since we feel it indicates more of the mathematical structure which leads to the development of new methods based on nonstandard inner products.

We emphasize that $\langle\cdot,\cdot\rangle$, $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ must be symmetric bilinear forms here, but we do not require them to be inner products for the theory presented in this section. For practical reasons, we will consider positivity/nonpositivity of symmetric bilinear forms and positive definiteness/indefiniteness of self-adjoint matrices separately from our considerations of symmetry and self-adjointness. Whenever we write $\langle\cdot,\cdot\rangle_{\mathcal{H}}$, $\mathcal{H}$ will be symmetric.

It is easy to verify the following lemmas.

LEMMA 3.1. [illegible] $\mathcal{A}_1$ [illegible] $\mathcal{A}_2$ [illegible] $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ [illegible] $\alpha, \beta \in \mathbb{R}$, $\alpha\mathcal{A}_1 + \beta\mathcal{A}_2$ [illegible] $\langle\cdot,\cdot\rangle_{\mathcal{H}}$

LEMMA 3.2. [illegible] $\mathcal{A}$ [illegible] $\langle\cdot,\cdot\rangle_{\mathcal{H}_1}$ [illegible] $\langle\cdot,\cdot\rangle_{\mathcal{H}_2}$ [illegible] $\mathcal{A}$ [illegible] $\langle\cdot,\cdot\rangle_{\alpha\mathcal{H}_1+\beta\mathcal{H}_2}$ [illegible] $\alpha, \beta \in \mathbb{R}$

Now if $\mathcal{A}$ is preconditioned on the left by $\mathcal{P}$, then from (3.3), $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ if and only if

$$(\mathcal{P}^{-1}\mathcal{A})^T\mathcal{H} = \mathcal{H}\mathcal{P}^{-1}\mathcal{A} \tag{3.4}$$

which is

$$\mathcal{A}^T\mathcal{P}^{-T}\mathcal{H} = \mathcal{H}\mathcal{P}^{-1}\mathcal{A}$$

or

$$\mathcal{A}^T(\mathcal{P}^{-T}\mathcal{H}) = (\mathcal{P}^{-T}\mathcal{H})^T\mathcal{A}$$

since $\mathcal{H}$ is symmetric. Thus if $\mathcal{A}$ is also symmetric we get

$$(\mathcal{P}^{-T}\mathcal{H})^T\mathcal{A} = \mathcal{A}(\mathcal{P}^{-T}\mathcal{H}) \tag{3.5}$$

and so we have the following lemma.

LEMMA 3.3. [illegible] $\mathcal{A}$, $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$ [illegible] $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ [illegible] $\mathcal{P}^{-T}\mathcal{H}$ [illegible] $\langle\cdot,\cdot\rangle_{\mathcal{A}}$

[illegible]. The proof follows directly from the above lemmas and (3.3). □

[illegible] 3.4. Lemma 3.3 includes the even more simple situations that $\mathcal{P}^{-1}\mathcal{A}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{P}}$ and $\mathcal{A}\mathcal{P}^{-1}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{A}^{-1}}$ when both $\mathcal{A}$ and $\mathcal{P}$ are

symmetric since I is trivially self-adjoint in any symmetric bilinear form. Clearly invertibility of $\mathcal{P}$ and $\mathcal{A}$, respectively, are needed in these two cases.

Now for symmetric $\mathcal{A}$, if $\mathcal{P}_1$ and $\mathcal{P}_2$ are such that $\mathcal{P}_i^{-1}\mathcal{A}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}_i}$, $i = 1, 2$, for symmetric matrices $\mathcal{H}_1$, $\mathcal{H}_2$, then

$$(\mathcal{P}_1^{-1}\mathcal{A})^T\mathcal{H}_1 = \mathcal{H}_1(\mathcal{P}_1^{-1}\mathcal{A}) \quad \text{and} \quad (\mathcal{P}_2^{-1}\mathcal{A})^T\mathcal{H}_2 = \mathcal{H}_2(\mathcal{P}_2^{-1}\mathcal{A}). \tag{3.6}$$

Using Lemma 3.3, $\mathcal{P}_i^{-T}\mathcal{H}_i$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{A}}$ for $i = 1, 2$, and thus by Lemma 3.1

$$\alpha\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{P}_2^{-T}\mathcal{H}_2$$

is also self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{A}}$ for any $\alpha, \beta \in \mathbb{R}$. Now, if for some α, β we are able to decompose the matrix $(\alpha\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{P}_2^{-T}\mathcal{H}_2) = \mathcal{P}_3^{-T}\mathcal{H}_3$ for some symmetric matrix $\mathcal{H}_3$, then $\mathcal{P}_3^{-T}\mathcal{H}_3$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{A}}$, and a further application of Lemma 3.3 yields that $\mathcal{P}_3^{-1}\mathcal{A}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}_3}$. We have now proved the following lemma.

LEMMA 3.5. $\mathcal{P}_1$ $\mathcal{P}_2$ $\mathcal{A}$ $\mathcal{H}_1$ $\mathcal{H}_2$ $\mathcal{P}_1^{-1}\mathcal{A}$ $\langle\cdot,\cdot\rangle_{\mathcal{H}_1}$ $\mathcal{P}_2^{-1}\mathcal{A}$ $\langle\cdot,\cdot\rangle_{\mathcal{H}_2}$

$$\alpha\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{P}_2^{-T}\mathcal{H}_2 = \mathcal{P}_3^{-T}\mathcal{H}_3$$

$\mathcal{P}_3$ $\mathcal{H}_3$ $\mathcal{P}_3^{-1}\mathcal{A}$ $\langle\cdot,\cdot\rangle_{\mathcal{H}_3}$

Lemma 3.5 shows a possible way to generate new preconditioners for $\mathcal{A}$. In section 6 we show practical examples of its use.

The construction of $\mathcal{P}_3$, $\mathcal{H}_3$ in Lemma 3.5 also allows straightforward inheritance of positive definiteness—for this to be a useful property it is essential that $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ defines an inner product, i.e., that $\mathcal{H}$ is positive definite. It is trivial to construct examples of indefinite diagonal matrices $\mathcal{A}$ and $\mathcal{H}$ for which $\langle\mathcal{A}w, w\rangle_{\mathcal{H}} > 0$ for all nonzero w, but in order to be able to take advantage of positive definiteness, for example, by employing conjugate gradients, it is important that $\langle w, w\rangle_{\mathcal{H}} = w^T\mathcal{H}w > 0$ for all nonzero w.

LEMMA 3.6. 3.5 $\mathcal{P}_i^{-1}\mathcal{A}$ $\langle\cdot,\cdot\rangle_{\mathcal{H}_i}$ $i = 1, 2$ $\mathcal{P}_3^{-1}\mathcal{A}$ $\langle\cdot,\cdot\rangle_{\mathcal{H}_3}$ α β

. Positive definiteness of $\mathcal{P}^{-1}\mathcal{A}$ in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ means that

$$\langle\mathcal{P}^{-1}\mathcal{A}w, w\rangle_{\mathcal{H}} > 0 \quad \text{for} \quad w \neq 0,$$

i.e., that $w^T\mathcal{A}\mathcal{P}^{-T}\mathcal{H}w > 0$ so that $\mathcal{A}\mathcal{P}^{-T}\mathcal{H}$ is a symmetric matrix with all eigenvalues positive. Thus all of $\mathcal{A}\mathcal{P}_1^{-T}\mathcal{H}_1$ and $\mathcal{A}\mathcal{P}_2^{-T}\mathcal{H}_2$ is symmetric and positive definite, and it follows that

$$\alpha\mathcal{A}\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{A}\mathcal{P}_2^{-T}\mathcal{H}_2 = \mathcal{A}\mathcal{P}_3^{-T}\mathcal{H}_3$$

must also be symmetric and positive definite for at least the positive values of α and β. ∎

We comment that there will, in general, be some negative values of α or β for which $\mathcal{P}_3^{-1}\mathcal{A}$ remains positive definite but at least one of α and β needs to be positive in this case. The precise limits on the values that α and β can take whilst positive definiteness is preserved depend on the extreme eigenvalues of $\mathcal{A}\mathcal{P}_1^{-T}\mathcal{H}_1$ and $\mathcal{A}\mathcal{P}_2^{-T}\mathcal{H}_2$.

Unfortunately, even if $\mathcal{H}_1$ and $\mathcal{H}_2$ are positive definite there is no guarantee that $\mathcal{H}_3$ will be also.

We can also consider right preconditioning: If $\widehat{\mathcal{A}} = \mathcal{A}\mathcal{P}^{-1}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$, then

$$\mathcal{P}^{-T}\mathcal{A}^T\mathcal{H} = \mathcal{H}\mathcal{A}\mathcal{P}^{-1} \text{ which is } (\mathcal{P}^{-1})^T(\mathcal{A}^T\mathcal{H}) = (\mathcal{A}^T\mathcal{H})^T\mathcal{P}^{-1}. \tag{3.7}$$

Thus, we have the following lemma.

LEMMA 3.7. [illegible] $\mathcal{P}$ [illegible] $\widehat{\mathcal{A}} = \mathcal{A}\mathcal{P}^{-1}$ [illegible] $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ [illegible] $\mathcal{H}$ [illegible] $\mathcal{A}^T\mathcal{H}$ [illegible] $\langle\cdot,\cdot\rangle_{\mathcal{P}^{-1}}$

Lemma 3.7 shows that we could combine problem matrices and symmetric bilinear forms for the same preconditioner. This is obviously more of a theoretical than a practical result compared to obtaining new preconditioners for a given problem as in the case of left preconditioning noted earlier. One of the decompositions as $\mathcal{P}_3^{-T}\mathcal{H}_3$, that we introduce in section 6 will provide not only a symmetric inner product matrix but also a symmetric preconditioner and therefore fulfills the conditions of Lemma 3.7.

We now want to discuss very briefly the eigenvalues of matrices which are self-adjoint according to our definition which allows indefinite symmetric bilinear forms. Assume that $\mathcal{A}^T\mathcal{H} = \mathcal{H}\mathcal{A}$ holds and that (λ, v) is a given eigenpair of $\mathcal{A}$. Thus,

$$\mathcal{A}v = \lambda v, \qquad v \neq 0. \tag{3.8}$$

Multiplying (3.8) from the left by $v^*\mathcal{H}$, where v^* is the conjugate transpose of v, gives

$$v^*\mathcal{H}\mathcal{A}v = \lambda v^*\mathcal{H}v. \tag{3.9}$$

Notice that the left-hand side of (3.9) is real since $\mathcal{H}\mathcal{A}$ is real symmetric. On the right-hand side $v^*\mathcal{H}v$ is also real since $\mathcal{H}$ is real symmetric, therefore the eigenvalue λ must be real. Note that, a matrix $\mathcal{H}$ always exists such that $\mathcal{A}^T\mathcal{H} = \mathcal{H}\mathcal{A}$ since any matrix is similar to its transpose; see, for example, Chapter 3.2 in [27]. In the context of the above theory, the interesting candidates for $\mathcal{H}$ are the real symmetric matrices; a complex symmetric matrix $\mathcal{H}$ always exists such that $\mathcal{A}^T\mathcal{H} = \mathcal{H}\mathcal{A}$ (cf. [27, section 2.3]).

Note that the above arguments establish that there is no symmetric bilinear form in which $\mathcal{A}$ is self-adjoint unless $\mathcal{A}$ has real eigenvalues.

It is also known that for a real diagonalizable matrix $\mathcal{A}$ which has only real eigenvalues there always exist inner products in which $\mathcal{A}$ is self-adjoint.

LEMMA 3.8. [illegible] $\mathcal{A} = R^{-1}\Lambda R$ [illegible] $\mathcal{A}$ [illegible] Λ [illegible] $\mathcal{A}$ [illegible] $\langle\cdot,\cdot\rangle_{R^T\Theta R}$ [illegible] Θ [illegible]. The conditions (3.3) for self-adjointness of $\mathcal{A}$ in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ are

$$R^T\Lambda R^{-T}\mathcal{H} = \mathcal{H}R^{-1}\Lambda R$$

which are clearly satisfied for $\mathcal{H} = R^T\Theta R$ whenever Θ is diagonal because then Θ and Λ commute. Clearly $\mathcal{H}$ is positive definite whenever the diagonal entries of Θ are positive. □

We remark that this result is not of great use in practice since knowledge of the complete eigensystem of $\mathcal{A}$ is somewhat prohibitive.

4. Saddle point examples. The first example is the classical method by Bramble and Pasciak already mentioned in the introduction. The preconditioner is given by

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ B & -I \end{bmatrix} \quad \text{and} \quad \mathcal{P}^{-1} = \begin{bmatrix} A_0^{-1} & 0 \\ BA_0^{-1} & -I \end{bmatrix} \tag{4.1}$$

and the symmetric bilinear form defined by the matrix

$$\mathcal{H} = \begin{bmatrix} A - A_0 & 0 \\ 0 & I \end{bmatrix}. \tag{4.2}$$

There are also extensions to the classical Bramble–Pasciak case; see [34, 28, 42]. In [34], for example, a Schur complement preconditioner S_0 is introduced into $\mathcal{P}$ giving

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ B & -S_0 \end{bmatrix} \quad \text{and} \quad \mathcal{P}^{-1} = \begin{bmatrix} A_0^{-1} & 0 \\ S_0^{-1}BA_0^{-1} & -S_0^{-1} \end{bmatrix}; \tag{4.3}$$

under certain conditions positive definiteness of the preconditioned saddle point system can still be guaranteed in a nonstandard inner product similar to (4.2), i.e.,

$$\mathcal{H} = \begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix}. \tag{4.4}$$

A similar analysis was provided by Zulehner in 2002; see [48]. Zulehner considered a preconditioner of the form (4.3) for an inexact Uzawa method which under certain conditions can admit the usability of a CG acceleration; see [33] for the connection of CG and the inexact Uzawa algorithm as a Richardson iteration method.

In 2006, Benzi and Simoncini gave a further example; see [5], which is an extension of an earlier work by Fischer et al. (cf. [15]). Namely,

$$\mathcal{P} = \mathcal{P}^{-1} = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \tag{4.5}$$

and

$$\mathcal{H} = \begin{bmatrix} A - \gamma I & B^T \\ B & \gamma I \end{bmatrix}. \tag{4.6}$$

Recently, Liesen and Parlett made an extension to this result taking a nonzero matrix C in (1.1) into account; see [30, 31]. In the language used here, the preconditioner is again

$$\mathcal{P} = \mathcal{P}^{-1} = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \tag{4.7}$$

but the symmetric bilinear form is now defined by

$$\mathcal{H} = \begin{bmatrix} A - \gamma I & B^T \\ B & \gamma I - C \end{bmatrix}. \tag{4.8}$$

There are certain conditions which must be satisfied by the parameter γ in order to guarantee positive definiteness of $\mathcal{H}$ so that CG in the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ can be reliably employed; see [5, 30, 31].

Liesen and Parlett also showed in [31] that the matrix $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$ is self-adjoint in every bilinear form of the type $\mathcal{H}p(\widehat{\mathcal{A}})$, where $\mathcal{H} = \mathcal{P}$ and $p(\widehat{\mathcal{A}})$ is any real polynomial in $\widehat{\mathcal{A}}$. The proof is based on a technique introduced by Freund (cf. [16]), where the matrix $\mathcal{H}$ can be shifted from one side of the polynomial $p(\mathcal{A})$ to the other by successively using $\widehat{\mathcal{A}}^T\mathcal{H} = \mathcal{H}\widehat{\mathcal{A}}$; see also section 7.3 of this paper. Trivially, this observation holds for general $\mathcal{H}$ whenever the condition $\widehat{\mathcal{A}}^T\mathcal{H} = \mathcal{H}\widehat{\mathcal{A}}$ is satisfied for any real symmetric $\mathcal{H}$ and not just for the matrix $\mathcal{H} = \mathcal{P}$. Through the choice of the polynomial p, the approach presented by Liesen and Parlett provides a whole set of interesting bilinear forms that may give useful examples.

Another example was given in Zulehner [48] in the context of inexact Uzawa methods and in [40] by Schöberl and Zulehner where the saddle point problem with $C = 0$ is preconditioned by the constraint preconditioner

$$\mathcal{P} = \begin{bmatrix} A_0 & B^T \\ B & BA_0^{-1}B^T - \hat{S} \end{bmatrix}$$

with A_0 and $\hat{S}$ being symmetric and positive definite. Then the preconditioned matrix is self-adjoint in the inner product defined by

$$\mathcal{H} = \begin{bmatrix} A_0 - A & 0 \\ 0 & BA_0^{-1}B^T - \hat{S} \end{bmatrix}.$$

Another example using a constraint preconditioner was given by Dohrmann and Lehoucq in [8]. They consider the general saddle point problem given in (1.1) with the constraint preconditioner

$$\mathcal{P} = \begin{bmatrix} \hat{S}_A & B^T \\ B & \hat{C} \end{bmatrix},$$

where $\hat{S}_A$ is an approximation to a penalized primal Schur complement $S_A = A + B^T\hat{C}^{-1}B$ and a symmetric and positive definite $\hat{C}$. The inner product in which the preconditioned matrix is self-adjoint is given by

$$\mathcal{H} = \begin{bmatrix} S_A - \hat{S}_A & 0 \\ 0 & \hat{C} - C \end{bmatrix}.$$

5. The modified Bramble–Pasciak preconditioner. The original Bramble–Pasciak CG method requires that the matrix

$$\mathcal{H} = \begin{bmatrix} A - A_0 & 0 \\ 0 & I \end{bmatrix}$$

is positive definite. The obvious drawback of this method is the necessity to scale the matrix A_0 such that $A - A_0$ is positive definite. Usually an eigenvalue problem for $A_0^{-1}A$ or at least an eigenvalue estimation problem has to be solved which can be costly; see [24] for a survey of methods that could be applied.

In contrast, we introduce the preconditioner

$$\mathcal{P}^+ = \begin{bmatrix} A_0 & 0 \\ -B & S_0 \end{bmatrix} \quad \text{and} \quad \left(\mathcal{P}^+\right)^{-1} = \begin{bmatrix} A_0^{-1} & 0 \\ BA_0^{-1} & S_0^{-1} \end{bmatrix} \tag{5.1}$$

and obtain by left preconditioning with $\mathcal{P}^+$

$$\widehat{\mathcal{A}} = \left(\mathcal{P}^+\right)^{-1}\mathcal{A} = \begin{bmatrix} A_0^{-1}A & A_0^{-1}B^T \\ S_0^{-1}BA_0^{-1}A + S_0^{-1}B & S_0^{-1}BA_0^{-1}B^T - S_0^{-1}C \end{bmatrix}. \tag{5.2}$$

Simple algebra shows that $\widehat{\mathcal{A}}$ is self-adjoint in the inner product induced by

$$\mathcal{H}^+ = \begin{bmatrix} A + A_0 & 0 \\ 0 & S_0 \end{bmatrix}, \tag{5.3}$$

where S_0 is a symmetric and positive definite Schur-complement preconditioner. An inner product of similar form to (5.3) was used by Zulehner in [48] in the analysis of inexact Uzawa methods. Note that for a positive definite preconditioner A_0 the matrix $\mathcal{H}^+$ is always positive definite due to the positive definiteness of the matrices A, A_0, and S_0. Thus, we are in this case always with an inner product and not just a symmetric bilinear form—whatever symmetric and positive definite A_0 is chosen—and so the appropriate Krylov subspace method can be used in this inner product. We will discuss the eigenvalue properties of the preconditioned matrix in section 8.

For the classical Bramble–Pasciak method (1.2), (1.3), Klawonn shows in [28] that the matrix

$$\widehat{\mathcal{A}}^T\mathcal{H} = \begin{bmatrix} AA_0^{-1}A - A & AA_0^{-1}B^T - B^T \\ BA_0^{-1}A - B & BA_0^{-1}B^T + C \end{bmatrix} \tag{5.4}$$

can be factorized as

$$\begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} AA_0^{-1}A - A & 0 \\ 0 & BA^{-1}B^T + C \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix}, \tag{5.5}$$

which is a congruence transformation. Now note that we can rewrite $AA_0^{-1}A - A$ as

$$A(A_0^{-1} - A^{-1})A,$$

which will be positive definite if $A_0^{-1} - A^{-1}$ is positive definite or, equivalently,

$$y^T A_0 y < y^T A y. \tag{5.6}$$

The condition (5.6) is precisely that required for $\mathcal{H}$ to be positive definite in this case. Since $BA^{-1}B^T + C$ is positive definite, Sylvester's law of inertia applied to (5.5) guarantees that $\widehat{\mathcal{A}}^T\mathcal{H}$ is positive definite, i.e., that $\widehat{\mathcal{A}}$ is self-adjoint and positive definite in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$.

The same has to be shown for the new preconditioner $\mathcal{P}^+$. Using Klawonn's approach we get for the matrix

$$\widehat{\mathcal{A}}^T\mathcal{H}^+ = \begin{bmatrix} AA_0^{-1}A + A & AA_0^{-1}B^T + B^T \\ BA_0^{-1}A + B & BA_0^{-1}B^T - C \end{bmatrix} \tag{5.7}$$

the decomposition

$$\begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} AA_0^{-1}A + A & 0 \\ 0 & -BA^{-1}B^T - C \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix}. \tag{5.8}$$

This shows that $\widehat{\mathcal{A}}^T\mathcal{H}^+$ is indefinite since $-BA^{-1}B^T - C$ is always negative definite and $AA_0^{-1}A + A$ is positive definite. Therefore the reliable applicability of the CG

method cannot be guaranteed. One alternative is to use a special implementation of MINRES which will be introduced in section 7.2.

We want to mention that the Bramble–Pasciak$^+$ preconditioner can also be interpreted as the classical Bramble–Pasciak preconditioner applied to the matrix $\mathcal{J}\mathcal{A}$, where $\mathcal{J} = \mathrm{diag}(I_n, -I_m)$ with I_j the identity of dimension $j = \{m, n\}$.

6. Examples of combination preconditioning.

6.1. Bramble–Pasciak combination preconditioning. Using Lemma 3.5 we want to analyze the possibility of combining the classical Bramble–Pasciak configuration with the Bramble–Pasciak$^+$ preconditioner introduced in the last section. Therefore, we have the preconditioners

$$\mathcal{P}_1 = \begin{bmatrix} A_0 & 0 \\ B & -I \end{bmatrix} \quad \text{and} \quad \mathcal{P}_2 = \begin{bmatrix} A_0 & 0 \\ -B & I \end{bmatrix}$$

and for the inner products

$$\mathcal{H}_1 = \begin{bmatrix} A - A_0 & 0 \\ 0 & I \end{bmatrix} \quad \text{and} \quad \mathcal{H}_2 = \begin{bmatrix} A + A_0 & 0 \\ 0 & I \end{bmatrix}.$$

Instead of $\alpha, \beta \in \mathbb{R}$ we use the combination parameters α and $1 - \alpha$ and get

$$\alpha \mathcal{P}_1^{-T} \mathcal{H}_1 + (1-\alpha)\mathcal{P}_2^{-T} \mathcal{H}_2 = \begin{bmatrix} A_0^{-1}A + (1-2\alpha)I & A_0^{-1}B^T \\ 0 & (1-2\alpha)I \end{bmatrix}.$$

If we find a decomposition as described in Lemma 3.5, then a new preconditioner and bilinear form are given. One factorization possibility would be

$$P_3^{-T} = \begin{bmatrix} A_0^{-1} & A_0^{-1}B^T \\ 0 & (1-2\alpha)I \end{bmatrix} \Longrightarrow P_3 = \begin{bmatrix} A_0 & 0 \\ \frac{1}{(2\alpha-1)}B & \frac{1}{1-2\alpha}I \end{bmatrix}$$

as the new preconditioner, and the bilinear form is then defined by

$$\mathcal{H}_3 = \begin{bmatrix} A + (1-2\alpha)A_0 & 0 \\ 0 & I \end{bmatrix}.$$

Note that, for $\alpha = 1$ we obtain the classical Bramble–Pasciak configuration, and $\alpha = 0$ gives the Bramble–Pasciak$^+$ setup. The obtained preconditioner can also be viewed as a special instance of an inexact Uzawa preconditioner; see [48].

We now have to analyze if positivity in the new bilinear form can be achieved and if the bilinear form is an inner product which can be exploited for short-term recurrence methods. Hence, the matrix

$$\widehat{\mathcal{A}}^T \mathcal{H}_3$$

with $\widehat{\mathcal{A}} = \mathcal{P}_3^{-1}\mathcal{A}$ has to be analyzed. The matrix

$$\widehat{\mathcal{A}}^T \mathcal{H}_3 = \begin{bmatrix} AA_0^{-1}A + (1-2\alpha)A & AA_0^{-1}B^T + (1-2\alpha)B^T \\ BA_0^{-1}A + (1-2\alpha)B & BA_0^{-1}B^T - (1-2\alpha)C \end{bmatrix}$$

can, similarly to the above, be factorized as the congruence transform

$$\widehat{\mathcal{A}}^T \mathcal{H}_3 =$$

$$\begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} AA_0^{-1}A + (1-2\alpha)A & 0 \\ 0 & (2\alpha-1)BA^{-1}B^T - (1-2\alpha)C \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix}.$$

The Sylvester law of inertia indicates that the number of positive and negative eigenvalues is determined by the eigenvalues of the matrix

$$\begin{bmatrix} AA_0^{-1}A + (1-2\alpha)A & 0 \\ 0 & (2\alpha-1)(BA_0^{-1}B^T + C) \end{bmatrix}$$

which we can analyze in a similar manner to (5.4), (5.5). We first consider the case where $C = 0$, and then it is easy to see that the block $(2\alpha - 1)BA_0^{-1}B^T$ is positive for $\alpha > 1/2$. With this choice for α we have to find conditions such that the block $AA_0^{-1}A + (1-2\alpha)A$ is also positive definite. Similar to the analysis made in section 7.1, we note the equivalence

$$A\left(A_0^{-1} + (1-2\alpha)A^{-1}\right)A,$$

and therefore positivity is given if $y^T A_0 y < (2\alpha - 1)y^T Ay$ which can also be written as

$$A_0 < (2\alpha - 1)A.$$

In addition, we want the matrix $\mathcal{H}_3$ to define an inner product which will be satisfied if the block $A + (1-2\alpha)A_0 > 0$, which is equivalent to

$$\frac{1}{2\alpha - 1}A > A_0.$$

Again, the case $\alpha = 1$ gives the Bramble–Pasciak configuration, and $\alpha = 0$ shows that there is no configuration that makes the Bramble–Pasciak$^+$ setup positive definite and CG reliably applicable. It is still possible to obtain a reliable CG method in the combination preconditioning case, i.e., if

$$A_0 < \min\left\{(2\alpha - 1)A, \frac{1}{2\alpha - 1}A\right\}$$

which is a more general restriction on A_0 than the Bramble–Pasciak case, $\alpha = 1$. The case $C \neq 0$ can be treated similarly since the block $(2\alpha - 1)(BA_0^{-1}B^T + C)$ will be positive for all $\alpha > 1/2$ and the above analysis applies.

6.2. Bramble–Pasciak and Benzi–Simoncini. As another less practical example which nevertheless shows how even very different methods can be combined, we consider $\mathcal{P}_1, \mathcal{H}_1$ defined by the classical Bramble–Pasciak method (4.1), (4.2) and $\mathcal{P}_2, \mathcal{H}_2$ defined by the Benzi–Simoncini approach (4.5), (4.6). From Lemma 3.5 we get

$$(\alpha\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{P}_2^{-T}\mathcal{H}_2) = \begin{bmatrix} (\alpha A_0^{-1} + \beta I)A - (\alpha + \beta\gamma)I & (\alpha A_0^{-1} + \beta I)B^T \\ -\beta B & -(\alpha + \beta\gamma)I \end{bmatrix}, \tag{6.1}$$

which is self-adjoint for all $\alpha, \beta \in \mathbb{R}$ in $\langle\cdot,\cdot\rangle_{\mathcal{A}}$. If we are able to split this into a new preconditioner $\mathcal{P}_3$ and a symmetric matrix $\mathcal{H}_3$, Lemma 3.5 guarantees that $\mathcal{P}_3^{-1}\mathcal{A}$ will be self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}_3}$.

One possibility is that

$$\mathcal{P}_3^{-T} = \begin{bmatrix} \alpha A_0^{-1} + \beta I & 0 \\ 0 & -\beta I \end{bmatrix} \text{ and } \mathcal{H}_3 = \begin{bmatrix} A - (\alpha + \beta\gamma)(\alpha A_0^{-1} + \beta I)^{-1} & B^T \\ B & \frac{\alpha+\beta\gamma}{\beta}I \end{bmatrix}. \tag{6.2}$$

Numerical results we have computed with this combination were less promising and we have omitted them. The bilinear form $\langle\cdot,\cdot\rangle_{\mathcal{H}_3}$ is not so convenient to work with.

6.3. Bramble–Pasciak and Schöberl–Zulehner. We now combine the Bramble–Pasciak CG and the method proposed by Schöberl and Zulehner in [40]. Therefore, we consider the preconditioners

$$\mathcal{P}_1 = \begin{bmatrix} A_0 & 0 \\ B & -S_0 \end{bmatrix} \quad \text{and} \quad \mathcal{P}_2 = \begin{bmatrix} A_0 & B^T \\ B & BA_0^{-1}B^T - \hat{S} \end{bmatrix}$$

and the inner products

$$\mathcal{H}_1 = \begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix} \quad \text{and} \quad \mathcal{H}_2 = \begin{bmatrix} A_0 - A & 0 \\ 0 & BA_0^{-1}B^T - \hat{S} \end{bmatrix}.$$

Again, we are looking for a factorization of $\alpha\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{P}_2^{-T}\mathcal{H}_2$ as $\mathcal{P}_3^{-T}\mathcal{H}_3$. Setting $S_0 = BA_0^{-1}B^T - \hat{S}$ yields

$$\begin{aligned} \alpha\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{P}_2^{-T}\mathcal{H}_2 &= \alpha \begin{bmatrix} A_0^{-1} & A_0^{-1}B^TS_0^{-1} \\ 0 & -S_0^{-1} \end{bmatrix} \begin{bmatrix} A - A_0 & \\ & S_0 \end{bmatrix} \\ &\quad + \beta \begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} A_0^{-1} & 0 \\ \hat{S}^{-1}BA_0^{-1} & -\hat{S}^{-1} \end{bmatrix} \begin{bmatrix} A_0 - A & \\ & S_0 \end{bmatrix}, \end{aligned} \tag{6.3}$$

which can be reformulated using

$$\begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_0 - A & 0 \\ 0 & S_0 \end{bmatrix} = \begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix} \tag{6.4}$$

and

$$\begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} -A_0^{-1} & 0 \\ 0 & -S_0^{-1} \end{bmatrix} = \begin{bmatrix} -A_0^{-1} & A_0^{-1}B^TS_0^{-1} \\ 0 & -S_0^{-1} \end{bmatrix}. \tag{6.5}$$

Hence, (6.3) simplifies to

$$\begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I \end{bmatrix} \left(\alpha \begin{bmatrix} -A_0^{-1} & 0 \\ 0 & -S_0^{-1} \end{bmatrix} + \beta \begin{bmatrix} A_0^{-1} & 0 \\ \hat{S}^{-1}BA_0^{-1} & -\hat{S}^{-1} \end{bmatrix} \right) \begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix}$$

with

$$\mathcal{P}_3^{-1} = \begin{bmatrix} (\beta - \alpha)A_0^{-1} & \beta A_0^{-1}B^T\hat{S}^{-1} \\ 0 & -(\alpha S_0^{-1} + \beta\hat{S}^{-1}) \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA_0^{-1} & I \end{bmatrix}$$

as the inverse of the new preconditioner, and as an inner product matrix we get

$$\mathcal{H}_3 = \begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix}.$$

The block $-(\alpha S_0^{-1} + \beta\hat{S}^{-1})$ of $\mathcal{P}_3^{-1}$ is not well suited for numerical purposes and we therefore try a different approach combining the Schöberl–Zulehner method with the Bramble–Pasciak CG. Thus, we consider the preconditioners

$$\mathcal{P}_1 = \begin{bmatrix} A_0 & 0 \\ B & -\hat{S} \end{bmatrix} \quad \text{and} \quad \mathcal{P}_2 = \begin{bmatrix} A_0 & B^T \\ B & BA_0^{-1}B^T - \hat{S} \end{bmatrix}$$

and the inner products

$$\mathcal{H}_1 = \begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S} \end{bmatrix} \quad \text{and} \quad \mathcal{H}_2 = \begin{bmatrix} A_0 - A & 0 \\ 0 & BA_0^{-1}B^T - \hat{S} \end{bmatrix}.$$

Once more, we try to find a decomposition as $\mathcal{P}_3^{-T}\mathcal{H}_3$ of

$$\begin{aligned} \alpha\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{P}_2^{-T}\mathcal{H}_2 &= \alpha \begin{bmatrix} A_0^{-1} & A_0^{-1}B^T\hat{S}^{-1} \\ 0 & -\hat{S}^{-1} \end{bmatrix} \begin{bmatrix} A - A_0 & \\ & \hat{S} \end{bmatrix} \\ &+ \beta \begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} A_0^{-1} & 0 \\ \hat{S}^{-1}BA_0^{-1} & -\hat{S}^{-1} \end{bmatrix} \begin{bmatrix} A_0 - A & \\ & BA_0^{-1}B^T - \hat{S} \end{bmatrix}. \end{aligned} \tag{6.6}$$

Using a simple modification of (6.4) then gives for (6.6)

$$\begin{aligned} \Bigg(\alpha \begin{bmatrix} A_0^{-1} & A_0^{-1}B^T\hat{S}^{-1} \\ 0 & -\hat{S}^{-1} \end{bmatrix} + \beta \begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} A_0^{-1} & 0 \\ \hat{S}^{-1}BA_0^{-1} & -\hat{S}^{-1} \end{bmatrix} \\ \begin{bmatrix} -I & \\ & (BA_0^{-1}B^T - \hat{S})\hat{S}^{-1} \end{bmatrix}\Bigg) \begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S} \end{bmatrix}. \end{aligned} \tag{6.7}$$

This can be further simplified using a modification of (6.5) for which the result is

$$\begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} (\alpha - \beta)A_0^{-1} & 0 \\ -\beta\hat{S}^{-1}BA_0^{-1} & (\beta - \alpha)\hat{S}^{-1} + \beta\hat{S}^{-1}BA_0^{-1}B^T\hat{S}^{-1} \end{bmatrix} \begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S} \end{bmatrix}.$$

The preconditioner is then given by

$$\mathcal{P}_3^{-1} = \begin{bmatrix} (\alpha - \beta)A_0^{-1} & -\beta A_0^{-1}B^T\hat{S}^{-1} \\ 0 & (\beta - \alpha)\hat{S}^{-1} - \beta\hat{S}^{-1}BA_0^{-1}B^T\hat{S}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA_0^{-1} & I \end{bmatrix} \tag{6.8}$$

with

$$\mathcal{H}_3 = \begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S} \end{bmatrix} \tag{6.9}$$

defining the bilinear form. It is also possible to reformulate the preconditioner presented by Schöberl and Zulehner using (6.8), i.e., $\beta = 1$ and $\alpha = 0$.

The method generated by combination preconditioning has a slightly more expensive preconditioner (6.8), i.e., one additional solve with $\hat{S}$ but the inner product matrix (6.9) is less expensive to evaluate because there is no need to solve with A_0. We assume here that $\hat{S}$ and A_0 are explicitly given which might not be the case when working with multigrid preconditioning for example.

7. Methods for solving the $\mathcal{P}^+$-preconditioned system.

7.1. The conjugate gradients method. The Hestenes and Stiefel [26] conjugate gradient method in an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is given in Algorithm 1; alternatively, see Chapter 2.1 in [11].

Algorithm 1 computes iterates such that $\|x - x_k\|_{\mathcal{H}\widehat{\mathcal{A}}}$ is minimal; for more details, see also Theorem 2.2 in [11]. It is clear that $\|\cdot\|_{\mathcal{H}\widehat{\mathcal{A}}}$ defines a norm when $\mathcal{H}\widehat{\mathcal{A}}$ is symmetric and positive definite. Algorithm 1 can be reliably applied whenever $\mathcal{H}$ and $\mathcal{H}\widehat{\mathcal{A}}$ are positive definite.

ALGORITHM 1 ALGORITHM FOR CG WITH INNER PRODUCT $\langle \cdot, \cdot \rangle_{\mathcal{H}}$.

Compute $r_0 = b - \widehat{\mathcal{A}} x_0$
Set $p_0 = r_0$
for $k = 1, 2, \ldots,$ **do**
  $\alpha_k = \langle r_k, r_k \rangle_{\mathcal{H}} / \langle \widehat{\mathcal{A}} p_k, p_k \rangle_{\mathcal{H}}$,
  $x_{k+1} = x_k + \alpha_k p_k$,
  $r_{k+1} = r_k - \alpha_k \widehat{\mathcal{A}} p_k$,
  $\beta_k = \langle r_{k+1}, r_{k+1} \rangle_{\mathcal{H}} / \langle r_k, r_k \rangle_{\mathcal{H}}$,
  $p_{k+1} = r_{k+1} + \beta_k p_k$,
end for

7.2. MINRES for the saddle point problem. In section 5 we showed that for positive definite preconditioners A_0 and S_0 the inner product matrix $\mathcal{H}^+$ will always be symmetric positive definite. Therefore, MINRES (minimal redisual method) introduced in 1975 by Paige and Saunders in [35], can be employed. It is typically the method of choice for symmetric indefinite systems. Since the preconditioned matrix $\widehat{\mathcal{A}}$ is symmetric in the inner product induced by $\mathcal{H}^+$, we can use a version of the classical Lanczos method to generate a basis for the Krylov subspace and then minimize the $\mathcal{H}^+$-norm of the residual. The $\mathcal{H}^+$-Lanczos method (cf. Algorithm 2) generates an $\mathcal{H}^+$-orthonormal basis for the Krylov subspace which can be expressed in matrix terms as

$$\widehat{\mathcal{A}} V_k = V_k T_{k,k} + \beta_k v_{k+1} e_k^T \tag{7.1}$$

with

$$T_{k,k} = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{bmatrix} \text{ and } V_k = [v_1, v_2, \ldots, v_k]$$

as well as $V_k^T \mathcal{H}^+ V_k = I$.

ALGORITHM 2 ALGORITHM FOR $\mathcal{H}^+$-LANCZOS.

Choose start vector $v_1 \mathbb{R}^n$ with $\|v_1\| = 1$.
Set $\beta_0 = 0$
for $k = 1, 2, \ldots,$ **do**
  $\tilde{v}_{k+1} = \widehat{\mathcal{A}} v_k - \beta_{k-1} v_{k-1}$,
  Compute $\alpha_k = \langle \tilde{v}_{k+1}, v_k \rangle_{\mathcal{H}^+}$
  $\tilde{v}_{k+1} = \tilde{v}_{k+1} - \alpha_k v_k$
  Set $\beta_k = \|\tilde{v}_{k+1}\|_{\mathcal{H}^+}$
  Set $v_{k+1} = \tilde{v}_{k+1} / \beta_k$
end for

Using the $\mathcal{H}^+$-Lanczos method it is easy to show that

$$\|r_k\|_{\mathcal{H}^+} = \| \|r_0\| \, e_1 - T_{k+1,k} y_k \|_{\mathcal{H}^+} \tag{7.2}$$

holds. Based on (7.2) we can implement a $\mathcal{H}^+$-MINRES process which minimizes the $\mathcal{H}^+$-norm of the residual (7.2) in complete analogy to the standard MINRES algorithm; details can be found in [35, 14, 25].

Applying the $\mathcal{H}^+$-MINRES method can also be justified by studying the problem in the context of the Faber–Manteuffel theorem. In 1984, Faber and Manteuffel [13] proved that only matrices which are normal(s) in some inner product admit an $(s+2)$-term recurrence method which minimizes some relevant quantity at each iteration step. In the most common case of 3-term recurrence methods such as CG or MINRES the normal(1) condition implies that the eigenvalues of the problem matrix lie on a straight line in the complex plane. A survey paper by Liesen and Strakoš (see [32]) gives a description of the Faber and Manteuffel paper in more accessible linear algebra terms. Recently, a new more elementary proof to the Faber–Manteuffel theorem was proposed in [12]. Essentially, a matrix M admits an $(s+2)$-term recurrence method if the G-adjoint,[1] $M^{\#} = GM^TG^{-1}$, can be expressed as a polynomial of degree s in M, i.e.,

$$M^{\#} = p_s(M).$$

With the choice of $M = \widehat{\mathcal{A}}$ and $G = (\mathcal{H}^+)^{-1}$ we get that for $\mathcal{H}^+$ and $\mathcal{P}^+$ a 3-term recurrence method always exists.

7.3. The simplified Lanczos method. The nonsymmetric Lanczos process (cf. [25, 38, 17, 20, 21]) generates two sequences of vectors v_k and w_k that are orthogonal to each other and are generated by

$$\rho_{k+1}v_{k+1} = \widehat{\mathcal{A}}v_k - \mu_k v_k - \nu_{k-1}v_{k-1} \tag{7.3}$$

for the first sequence and

$$\zeta_{k+1}w_{k+1} = \widehat{\mathcal{A}}^T w_k - \mu_k w_k - \tfrac{\nu_{k-1}\rho_k}{\zeta_k} w_{k-1} \tag{7.4}$$

for the second sequence with $\mu_k = w_k^T\widehat{\mathcal{A}}v_k/w_k^Tv_k$ and $\nu_k = \zeta_k w_k^T v_k/w_{k-1}^T v_{k-1}$. There is more than one way to scale the two vectors in every iteration step. Here, we use $\|v_j\| = 1$ and $\|w_j\| = 1$. The biorthogonality condition between W_k and V_k, i.e., $W_k^TV_k = D_k$, gives

$$D_k = \operatorname{diag}(\delta_1, \delta_2, \ldots, \delta_k), \text{ where } \delta_j = w_j^T v_j. \tag{7.5}$$

Furthermore, we can now write the recursions in terms of matrices and get

$$\widehat{\mathcal{A}}V_k = V_{k+1}T_{k+1,k} \tag{7.6}$$

as well as

$$\widehat{\mathcal{A}}^T W_k = W_{k+1}\Gamma_{k+1}^{-1}T_{k+1,k}\Gamma_{k+1}, \tag{7.7}$$

where the matrix Γ_k is defined as

$$\Gamma_k = \operatorname{diag}(\gamma_1, \gamma_2, \ldots, \gamma_k) \text{ with } \gamma_j = \begin{cases} 1 & \text{if } j = 1, \\ \gamma_{j-1}\rho_j/\zeta_j & \text{if } j > 1. \end{cases} \tag{7.8}$$

[1] G is a symmetric positive definite inner product matrix.

One advantage of the nonsymmetric Lanczos process is that $T_{k+1,k}$ is a tridiagonal matrix with $T_{k,k}$ typically nonsymmetric. There are different cases where the nonsymmetric Lanczos process can break down. The first case, the so-called [illegible], indicates that the solution lies already in the current Krylov space. In the case of $w_j^T v_j = 0$, neither v_{j+1} nor w_{j+1} are zero and the so-called [illegible] occurs. A remedy is to use look-ahead strategies; see [36, 18] for more details. For the so-called [illegible] the look-ahead strategies fail.

The nonsymmetric Lanczos process can now be simplified using the self-adjointness of $\widehat{\mathcal{A}}$ in the $\mathcal{H}^+$-inner product, i.e.,

$$\widehat{\mathcal{A}}^T \mathcal{H}^+ = \mathcal{H}^+ \widehat{\mathcal{A}}.$$

In [20] Freund and Nachtigal observe that for the Lanczos vectors the relations

$$v_j = \phi_j(\widehat{\mathcal{A}}) v_1 \quad \text{and} \quad w_j = \gamma_j \phi_j(\widehat{\mathcal{A}}^T) w_1 \tag{7.9}$$

hold, where ϕ is the so-called [illegible] which is of polynomial degree $j-1$. Using (7.9) and setting $w_1 = \mathcal{H}^+ v_1$, we get

$$w_j = \gamma_j \phi_j(\widehat{\mathcal{A}}^T) w_1 = \gamma_j \phi_j(\widehat{\mathcal{A}}^T) \mathcal{H}^+ v_1 = \gamma_j \mathcal{H}^+ \phi_j(\widehat{\mathcal{A}}) v_1 = \gamma_j \mathcal{H}^+ v_j.$$

Hence, we can compute the vector w_j without multiplying by $\widehat{\mathcal{A}}^T$. Instead,

$$w_{j+1} = \gamma_{j+1} \mathcal{H}^+ v_{j+1} \tag{7.10}$$

can be used. The parameter $\gamma_{j+1} = \gamma_j \rho_{j+1} / \zeta_{j+1}$ involves ζ_{j+1} which cannot be computed at that time. Thus the relation (7.10) has to be reformulated to

$$\tilde{w}_{j+1} = \zeta_{j+1} w_{j+1} = \gamma_j \rho_{j+1} \mathcal{H}^+ v_{j+1} = \gamma_j \mathcal{H}^+ \tilde{v}_{j+1}$$

which gives us now a computable version of the simplified Lanczos method; see Algorithm 3.

ALGORITHM 3 ALGORITHM FOR THE SIMPLIFIED LANCZOS METHOD.

Choose v_1 and compute $w_1 = \mathcal{H}^+ v_1$
Compute $\rho_1 = \|v_1\|$ and $\zeta_1 = \|w_1\|$
Set $\gamma_1 = \frac{\rho_1}{\zeta_1}$
for $k = 1, 2, \ldots,$ **do**
  Compute $\mu_k = (w_k^T \widehat{\mathcal{A}} v_k)/(w_k^T v_K)$
  Set $\nu_k = \zeta_k (w_k^T v_k)/(w_{k-1}^T v_{k-1})$
  $v_{k+1} = A v_k - \mu_k v_k - \nu_k v_{k-1}$
  $w_{k+1} = \gamma_k v_{k+1}$
  Compute $\rho_{k+1} = \|v_{k+1}\|$ and $\zeta_{k+1} = \|w_{k+1}\|$
  Set $\gamma_{k+1} = \gamma_k \rho_{k+1} / \zeta_{k+1}$
end for

7.4. The ideal transpose-free QMR method. In [17] Freund introduced the ideal transpose-free QMR method (ITFQMR) by using the simplification of the Lanczos method. Freund's implementation is based on a QMR-from-BICG procedure and coupled two-term recurrence relations; details can be found in [17, 21]. The method is also sometimes called SQMR or simplified QMR.

Here, we introduce the ITFQMR algorithm based on the simplified Lanczos method. In more detail, we have

$$\widehat{\mathcal{A}}V_k = V_{k+1}T_{k+1,k} \tag{7.11}$$

from the nonsymmetric Lanczos process and as a result get

$$r_k = V_{k+1}(\|r_0\| e_1 - T_{k+1,k}y_k) \tag{7.12}$$

for the residual. The term $(\|r_0\| e_1 - T_{k+1,k}y_k)$ is called the quasi-residual. A method based on minimizing the quasi-residual is QMR which was introduced in [19]. There the least squares problem $(\|r_0\| e_1 - T_{k+1,k}y_k)$ can be solved via an updated QR factorization that only requires one Givens rotation per step, a technique well known from MINRES [35].

The ITFQMR method is based on minimizing the quasi-residual in the same way QMR does but in the underlying nonsymmetric Lanczos process multiplications with the transpose are omitted and replaced by multiplications with the matrix $\mathcal{H}^+$. When using QMR and similar methods, we have to keep in mind that the quantities minimized here, the quasi-residual in the case of ITFQMR, are much less understood as the corresponding quantities used in MINRES and CG. Furthermore, as a method based on the nonsymmetric Lanczos, ITFQMR can break down and look-ahead strategies have to be employed; see [36, 18] for more details. There are also incurable breakdowns but from our experience it is hard to find them in practical applications.

8. Eigenvalue analysis. For a better understanding of the convergence behavior of the presented methods, we analyze the eigenvalues of $(\mathcal{P}^+)^{-1}\mathcal{A}$ by looking at the generalized eigenvalue problem $\mathcal{A}v = \lambda\mathcal{P}^+v$, i.e.,

$$\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \lambda \begin{bmatrix} A_0 & 0 \\ -B & S_0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}. \tag{8.1}$$

From (8.1) we get

$$Av_1 + B^Tv_2 = \lambda A_0 v_1 \tag{8.2}$$

and

$$Bv_1 - C^Tv_2 = -\lambda Bv_1 + \lambda S_0 v_2. \tag{8.3}$$

We first analyze the case where $A_0 = A$ and for $\lambda = 1$ from (8.2) get that $B^Tv_2 = 0$, and therefore $v_2 = 0$ under the condition that $Bv_1 = 0$. Since the kernel of B is $(n-m)$-dimensional we have $\lambda = 1$ with multiplicity $n - m$. For the case $\lambda \neq 1$, (8.2) gives

$$v_1 = \frac{1}{\lambda - 1}A^{-1}B^Tv_2,$$

which we substitute into (8.3) to get

$$BA^{-1}B^Tv_2 = \frac{\lambda(\lambda-1)}{\lambda+1}S_0v_2 + \frac{\lambda-1}{\lambda+1}Cv_2.$$

For $C = 0$ the remaining $2m$ eigenvalues of the preconditioned matrix $\widehat{\mathcal{A}}$ are given by the eigenvalues σ of

$$S_0^{-1}BA^{-1}B^T$$

and the relation

$$\sigma = \frac{\lambda(\lambda - 1)}{\lambda + 1}.$$

Hence, the eigenvalues of $\widehat{\mathcal{A}}$ become

$$\lambda_{1,2} = \frac{1+\sigma}{2} \pm \sqrt{\frac{(1+\sigma)^2}{4} + \sigma}. \tag{8.4}$$

Obviously, $\sigma > 0$, and therefore we have m negative eigenvalues given by (8.4). This shows that there are $2m + 1$ different eigenvalues, and we expect the method to terminate in at most $2m+1$ steps. A similar analysis for the classical Bramble–Pasciak can be found in [42].

In contrast, the eigenvalues of the preconditioned saddle point problem $\mathcal{P}^{-1}\mathcal{A}$ in the case of $\mathcal{P}$ being the block diagonal preconditioner

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ 0 & S_0 \end{bmatrix}$$

with $A_0 = A$ and $C = 0$ are given by $n-m$ unit eigenvalues and again the eigenvalues σ of

$$S_0^{-1} B A^{-1} B^T$$

via the relation

$$\sigma = \lambda(\lambda - 1) \text{ with } \lambda_{1,2} = \frac{1}{2} \pm \sqrt{\frac{1}{4} + \sigma}.$$

Since we expect S_0 to be a good preconditioner for $BA^{-1}B^T$, we expect the eigenvalues to not differ too much from unit eigenvalues which would give a similar convergence for the block-diagonal and the Bramble–Pasciak$^+$ preconditioner. Figure 8.1 illustrates how the eigenvalues of the preconditioned matrix in the case of block-diagonal preconditioning (dashed line), and in the case of Bramble–Pasciak$^+$ preconditioning (solid line) depend on the eigenvalues σ of $BA^{-1}B^T$ in a region around 1.

The indefiniteness of $(\mathcal{P}^+)^{-1}\mathcal{A}$ indicates that methods such as $\mathcal{H}^+$-MINRES or ITFQMR should be used. We will illustrate their convergence behavior in section 9 by applying them to Stokes examples from the IFISS software [10].

Following the analysis presented in [42], we also want to analyze the case $A_0 \neq A$. Therefore, we consider the symmetric and positive-definite block-diagonal preconditioner

$$\mathcal{M} = \begin{bmatrix} A_0 & 0 \\ 0 & S_0 \end{bmatrix}$$

and the generalized eigenvalue problem $\mathcal{A}u = \lambda \mathcal{P}^+ u$. Using $v = \mathcal{M}^{1/2}u$ we get $\mathcal{M}^{-1/2}\mathcal{A}\mathcal{M}^{-1/2}v = \lambda \mathcal{M}^{-1/2}\mathcal{P}^+\mathcal{M}^{-1/2}v$. This gives rise to a new generalized eigenvalue problem $\tilde{\mathcal{A}}v = \lambda \tilde{P} v$ with

$$\tilde{\mathcal{A}} = \begin{bmatrix} A_0^{-1/2} A A_0^{-1/2} & A_0^{-1/2} B^T S_0^{-1/2} \\ S_0^{-1/2} B A_0^{-1/2} & -S_0^{-1/2} C S_0^{-1/2} \end{bmatrix} = \begin{bmatrix} \tilde{A} & \tilde{B}^T \\ \tilde{B} & -\tilde{C} \end{bmatrix}$$

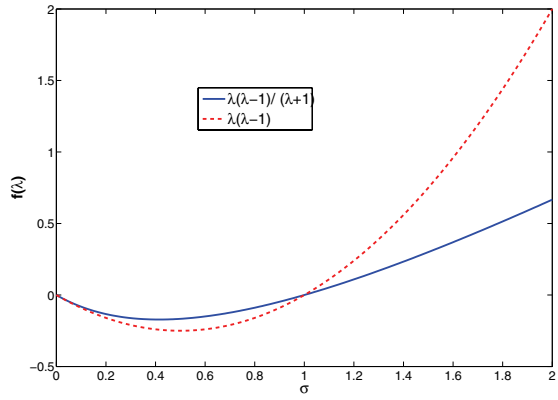


FIG. 8.1. *Dependence of eigenvalues on* σ.

and

$$\tilde{\mathcal{P}} = \begin{bmatrix} I & 0 \\ -S_0^{-1/2} B A_0^{-1/2} & I \end{bmatrix} = \begin{bmatrix} I & 0 \\ -\tilde{B} & I \end{bmatrix}.$$

The eigenvalue problem can hence be reformulated as

$$\tilde{A}v_1 + \tilde{B}^T v_2 = \lambda v_1, \tag{8.5}$$

$$\tilde{B}v_1 - \tilde{C}v_2 = -\lambda \tilde{B}v_1 + \lambda v_2. \tag{8.6}$$

Assume now that $v_2 = 0$ yields $\tilde{A}v_1 = \lambda v_1$ with λ an eigenvalue of the symmetric positive definite matrix $\tilde{A}$ if only $(1+\lambda)\tilde{B}v_1 = 0$. The case $v_1 = 0$ implies that $\tilde{B}^T v_2 = 0$, but since $\tilde{B}^T$ is of full rank $v_2 = 0$. Thus, we assume that $v_1 \neq 0$ and $v_2 \neq 0$. If we multiply (8.5) on the left by the conjugate transpose v_1^*, then we obtain

$$v_1^* \tilde{A} v_1 + v_1^* \tilde{B}^T v_2 = \lambda v_1^* v_1 \Longrightarrow v_1^* \tilde{B}^T v_2 = \lambda v_1^* v_1 - v_1^* \tilde{A} v_1. \tag{8.7}$$

The conjugate transpose of (8.6) multiplied on the right by v_2 gives

$$v_1^* \tilde{B}^T v_2 - v_2^* \tilde{C} v_2 = -\bar{\lambda} v_1^* \tilde{B} v_2 + \bar{\lambda} v_2^* v_2. \tag{8.8}$$

Using (8.7) for (8.8), we obtain

$$(1+\bar{\lambda})(\lambda v_1^* v_1 - v_1^* \tilde{A} v_1) - v_2^* \tilde{C} v_2 = \bar{\lambda} v_2^* v_2,$$

which can be further simplified to

$$(\lambda + |\lambda|^2) \|v_1\|^2 - (1+\bar{\lambda}) v_1^* \tilde{A} v_1 - v_2^* \tilde{C} v_2 - \bar{\lambda} \|v_2\|^2 = 0. \tag{8.9}$$

Assuming that $\lambda = a + ib$, we can analyze the imaginary part of (8.9) and obtain

$$b(\|v_1\|^2 + v_1^* \tilde{A} v_1 + \|v_2\|^2) = 0$$

which states that $b = 0$, and therefore all eigenvalues must be real. This underlines the argument made earlier about the use of short-term recurrence methods such as MINRES since all eigenvalues of the preconditioned matrix are on the real line.

We analyze (8.9) further knowing that λ is real and under the assumption that $\|v\| = 1$ with $\|v_2\|^2 = 1 - \|v_1\|^2$ and obtain

$$(\lambda + |\lambda|^2)\|v_1\|^2 - \lambda v_1^* \tilde{A} v_1 - \lambda + \lambda \|v_1\|^2 - v_1^* \tilde{A} v_1 - v_2^* \tilde{C} v_2 = 0. \tag{8.10}$$

We then get for λ

$$\lambda_\pm = \frac{v_1^* \tilde{A} v_1 + 1 - 2\|v_1\|^2}{2\|v_1\|^2} \pm \sqrt{\frac{(v_1^* \tilde{A} v_1 + 1 - 2\|v_1\|^2)^2}{4\|v_1\|^4} + \frac{v_1^* \tilde{A} v_1 + v_2^* \tilde{C} v_2}{\|v_1\|^2}}. \tag{8.11}$$

Note that, $v_1^* \tilde{A} v_1 + v_2^* \tilde{C} v_2 \geq 0$ for all v_1 and v_2. Since $\tilde{A}$ and $\tilde{C}$ are both symmetric matrices, we have the following bounds:

$$\mu_{min}^{\tilde{C}} \leq v_2^* \tilde{C} v_2 \leq \mu_{max}^{\tilde{C}}$$

and

$$\mu_{min}^{\tilde{A}} \leq v_1^* \tilde{A} v_1 \leq \mu_{max}^{\tilde{A}}$$

with $\mu_{min}^{\tilde{C}}$ and $\mu_{min}^{\tilde{A}}$ the minimal eigenvalue of $\tilde{C}$ and $\tilde{A}$, respectively, and $\mu_{max}^{\tilde{C}}$ and $\mu_{max}^{\tilde{A}}$ the maximal eigenvalue of $\tilde{C}$ and $\tilde{A}$, respectively. We first assume that $v_1^* \tilde{A} v_1 + 1 - 2\|v_1\|^2 \geq 0$ and get

$$\frac{\mu_{min}^{\tilde{A}} + 1 - 2\|v_1\|^2}{2\|v_1\|^2} + \sqrt{\frac{(\mu_{min}^{\tilde{A}} + 1 - 2\|v_1\|^2)^2}{4\|v_1\|^4} + \mu_{min}^{\tilde{A}} + \mu_{min}^{\tilde{C}}} \leq \lambda_+$$

and

$$\lambda_+ \leq \frac{\mu_{max}^{\tilde{A}} + 1 - 2\|v_1\|^2}{2\|v_1\|^2} + \sqrt{\frac{(\mu_{max}^{\tilde{A}} + 1 - 2\|v_1\|^2)^2}{4\|v_1\|^4} + \mu_{max}^{\tilde{A}} + \mu_{max}^{\tilde{C}}}$$

as well as

$$\frac{\mu_{min}^{\tilde{A}} + 1 - 2\|v_1\|^2}{2\|v_1\|^2} - \sqrt{\frac{(\mu_{max}^{\tilde{A}} + 1 - 2\|v_1\|^2)^2}{4\|v_1\|^4} + \mu_{max}^{\tilde{A}} + \mu_{max}^{\tilde{C}}} \leq \lambda_-$$

and

$$\lambda_+ \leq \frac{\mu_{max}^{\tilde{A}} + 1 - 2\|v_1\|^2}{2\|v_1\|^2} - \sqrt{\frac{(\mu_{min}^{\tilde{A}} + 1 - 2\|v_1\|^2)^2}{4\|v_1\|^4} + \mu_{min}^{\tilde{A}} + \mu_{min}^{\tilde{C}}}.$$

A similar analysis can be made for the case $v_1^* \tilde{A} v_1 + 1 - 2\|v_1\|^2 < 0$.

9. Numerical experiments. In this section we show the results of our numerical experiments. The matrices are coming from the Stokes problem and, in particular, were generated using the IFISS[2] package [10]. The Stokes problem is given by

$$\begin{aligned} -\nabla^2 u + \nabla p &= f, \\ \nabla \cdot u &= 0 \end{aligned} \tag{9.1}$$

[2]http://www.maths.manchester.ac.uk/~djs/ifiss/.

with appropriate boundary conditions which can be found in [11]. Namely, we consider the flow over the channel domain (Example 5.1.1 in [11]) and the flow over a backward facing step (Example 5.1.2 in [11]). Equation (9.1) can be transformed using a weak formulation which can then be treated using the finite element method; see [11] for details. The linear system governing the finite element method for the Stokes problem is a saddle point problem

$$\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix},$$

where $C \neq 0$ for stabilized elements. This matrix is symmetric but indefinite and could be treated with MINRES in the first place. But in order to improve the convergence we have to compare our Bramble–Pasciak$^+$ preconditioner to other suitable methods. One candidate would be the block-diagonal preconditioning introduced by Silvester and Wathen in [47, 41]. There a block-diagonal preconditioner

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ 0 & S_0 \end{bmatrix} \tag{9.2}$$

is used in which A_0 is a preconditioner for A and S_0 is a Schur complement preconditioner. We can use MINRES with this type of preconditioner; see [4] for details. In the Bramble–Pasciak$^+$ setup the preconditioned matrix is symmetric in the $\mathcal{H}^+$-inner product. This enables us to use the $\mathcal{H}^+$-MINRES method introduced in section 7.2. We will compare this method to the classical MINRES algorithm for the block-diagonal preconditioner. In the IFISS implementation the preconditioner S_0 is chosen to be the positive-definite pressure mass matrix; see Chapter 6.2 in [11]. The right-hand side for each example is also given by IFISS.

9.1. The first example considered is based on the flow over a backward facing step. The size of the system matrix $\mathcal{A}$ is given by 6659×6659 with $m = 769$ and $n = 5890$. The results shown in Figure 9.1 are obtained by using the $\mathcal{H}^+$-MINRES method and the classical preconditioned MINRES as given in [47, 41] as well as CG for the classical Bramble–Pasciak setup. The preconditioner A_0 is given by the incomplete Cholesky factorization. In particular, we use MATLAB's implementation with no additional fill-in; see [38] for details. S_0 is given by IFISS as the pressure mass matrix. The blue (dashed) curve is showing the results of the preconditioned MINRES with a block-diagonal preconditioner. The corresponding preconditioned residual is given in the 2-norm. The black (dash-dotted) line shows the 2-norm preconditioned residuals computed by the $\mathcal{H}^+$-MINRES algorithm. The red (solid) curve shows the preconditioned residuals for CG with the Bramble–Pasciak setup. As expected from the eigenvalue analysis in section 8, the results for MINRES and $\mathcal{H}^+$-MINRES are very similar and are both outperformed by the Bramble–Pasciak CG except for rather large convergence tolerances.

9.2. This example comes again from IFISS and is representing the flow of a channel domain. The size of the system matrix $\mathcal{A}$ is given by 9539×9539 with $m = 1089$ and $n = 8450$. The results shown in Figure 9.2 are obtained by using the $\mathcal{H}^+$-MINRES method and the classical preconditioned MINRES, as given in [47, 41], as well as ITFQMR for the classical Bramble–Pasciak setup. The preconditioners are chosen such that $A_0 = A$ and S_0 is given by IFISS as the pressure mass matrix. The blue (dashed) curve is showing the results of the preconditioned MINRES with a block-diagonal preconditioner. The corresponding preconditioned residual is given in the 2-norm. The black (dash-dotted) line shows the 2-norm preconditioned residuals

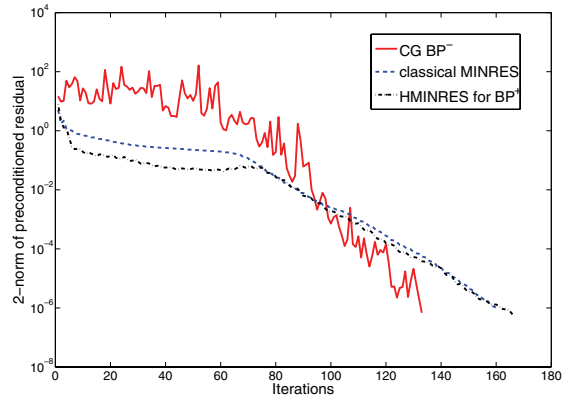


FIG. 9.1. *Results for* $\mathcal{H}^+$-MINRES, *classical preconditioned* MINRES, *and* CG *for classical Bramble–Pasciak.*

computed by the $\mathcal{H}^+$-MINRES algorithm. The red (solid) curve shows the preconditioned residuals for ITFQMR with the Bramble–Pasciak setup. Again, the results for MINRES and $\mathcal{H}^+$-MINRES are very similar.

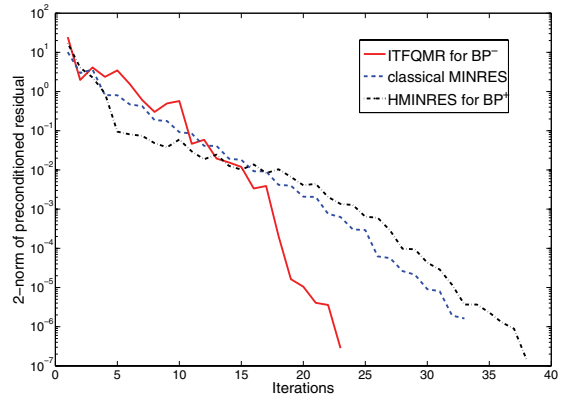


FIG. 9.2. *Results for* $\mathcal{H}^+$-MINRES, *classical preconditioned* MINRES, *and* ITFQMR *for classical Bramble–Pasciak.*

We now show results for the combination preconditioning with the Bramble–Pasciak and the Bramble–Pasciak$^+$ setup.

9.3. In this example the matrix represents the flow over a channel domain and is of size 9539×9539. Our choice for A_0 is again the incomplete Cholesky decomposition with zero fill-in and S_0 the pressure mass matrix. Figure 9.3 shows the results for different values of α. The choice for $\alpha = 2/3$ shown in the black (solid) curve performs better than the original Bramble–Pasciak method reflected by $\alpha = 1$ in the blue (dashed) line. For comparison we also show the results for the preconditioned MINRES in the red (dashed) line. Further values of α are shown.

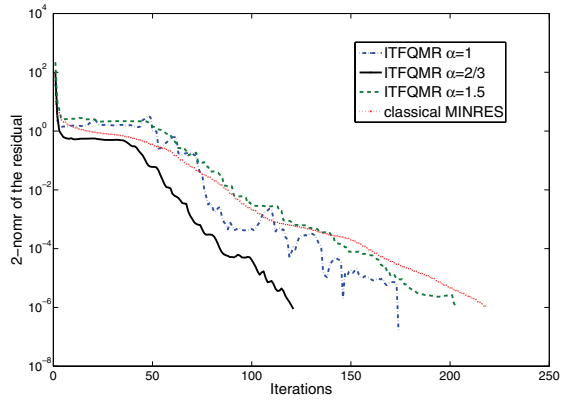


FIG. 9.3. ITFQMR *results for combination preconditioning with different values for* α.

EXAMPLE 9.4. The setup for this example is identical to the one described in Example 9.3, only the underlying matrix now comes from the flow over the backward facing step. The dimension of $\mathcal{A}$ is 6659×6659. Again for $\alpha = 2/3$, the combination preconditioning outperforms ITFQMR with the Bramble–Pasciak setup; see Figure 9.4.

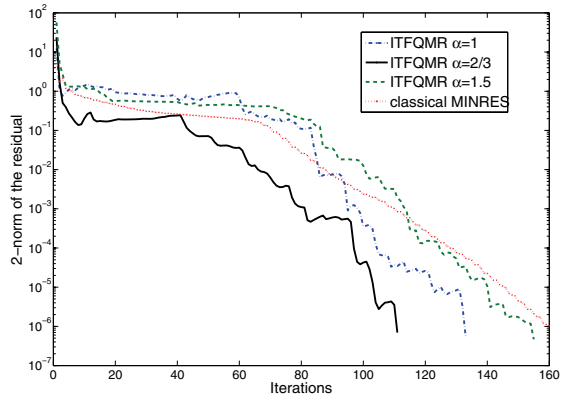


FIG. 9.4. ITFQMR *results for combination preconditioning with different values for* α.

The combination of the Bramble–Pasciak setup and the method of Schöberl and Zulehner as presented in section 6.3 is given by the preconditioner

$$\mathcal{P}_3^{-1} = \begin{bmatrix} (\alpha-\beta)A_0^{-1} & -\beta A_0^{-1}B^T\hat{S}^{-1} \\ 0 & (\beta-\alpha)\hat{S}^{-1} - \beta\hat{S}^{-1}BA_0^{-1}B^T\hat{S}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA_0^{-1} & I \end{bmatrix}$$

and inner product

$$\mathcal{H}_3 = \begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S} \end{bmatrix}.$$

EXAMPLE 9.5. In this example we apply CG with the combination preconditioning setup for Schöberl–Zulehner and Bramble–Pasciak to a linear system coming from

the flow over a backward facing step of dimension 6659 as shown in Figure 9.5. The preconditioner A_0 is chosen to be the zero fill-in incomplete Cholesky factorization and $\hat{S}$ is the pressure mass matrix given in IFISS. For the parameter choice $\alpha = -.3$ and $\beta = .5$ the combination is able to outperform the method of Schöberl and Zulehner.

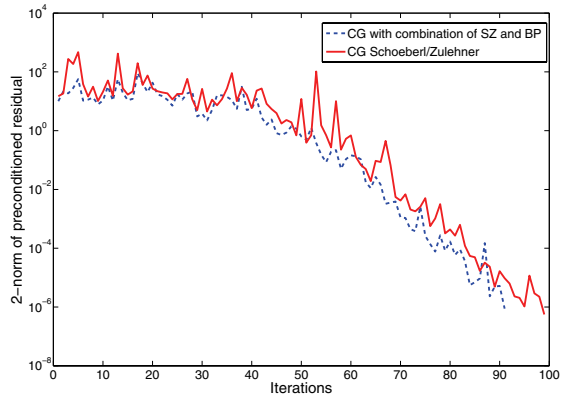


FIG. 9.5. *CG for Schöberl–Zulehner and combination preconditioning for* $\alpha = -.3$ *and* $\beta = .5$.

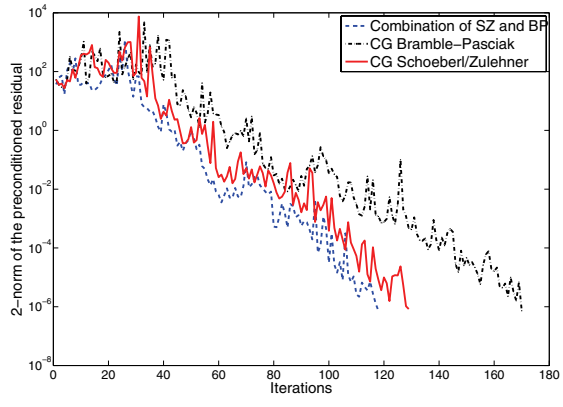


FIG. 9.6. *CG for Schöberl–Zulehner, Bramble–Pasciak, and combination preconditioning for* $\alpha = -.3$ *and* $\beta = .5$.

9.6. In this example we apply CG with the combination preconditioning setup for Schöberl–Zulehner and Bramble–Pasciak to a linear system coming from the flow over the channel domain of dimension 9539. In addition, we also show the results for CG with the Bramble–Pasciak setup; see Figure 9.6. The preconditioner A_0 is chosen to be the zero fill-in incomplete Cholesky factorization and $\hat{S}$ is the pressure mass matrix given in IFISS. For the parameter choice $\alpha = -.3$ and $\beta = .5$ the combination is able to outperform the method of Schöberl and Zulehner.

10. Conclusions. We have explained the general concept of self-adjointness in nonstandard inner products or symmetric bilinear forms and, in the specific case of saddle point problems, have shown how a number of known examples fit into this

paradigm. We have indicated how self-adjointness may be taken advantage of in the choice of the iterative solution method of Krylov subspace type—in general, it is more desirable to be able to work with iterative methods for self-adjoint matrices rather than general nonsymmetric matrices because of the greater efficiency of symmetric iterative methods. The understanding of the convergence of symmetric iterative methods like CG is much more secure and descriptive than for nonsymmetric methods.

The possibility of combination preconditioning by exploiting self-adjointness in different nonstandard inner products or symmetric bilinear forms has been analyzed and examples given of how two methods can be combined to obtain a new preconditioner and a different symmetric bilinear form. The first example combines the new BP^+ method which we have introduced with the classical Bramble–Pasciak method. We demonstrate that a particular combination outperforms the widely used classical method; it requires fewer iterations while the work per iteration is the same. The second example is of more academic than practical value. The third example combines the BP method and a recently introduced method by Schöberl and Zulehner. The combination preconditioning method was able to outperform both the Bramble–Pasciak CG and Schöberl–Zulehner CG method.

Our analysis may provide the basis for the discovery of further useful examples where self-adjointness may hold in nonstandard inner products and also shows how preconditioning can usefully be employed to create rather than destroy symmetry of matrices.

Acknowledgment. The authors would like to thank the two anonymous referees for their careful reading and helpful comments.

REFERENCES

[1] M. Ainsworth and S. Sherwin, *Domain decomposition preconditioners for p and hp finite element approximation of Stokes equations*, Comput. Methods Appl. Mech. Engrg., 175 (1999), pp. 243–266.

[2] O. Axelsson and M. Neytcheva, *Preconditioning methods for linear systems arising in constrained optimization problems*, Numer. Linear Algebra Appl., 10 (2003), pp. 3–31.

[3] O. Axelsson and M. Neytcheva, *Robust preconditioners for saddle point problems*, in Numerical Methods and Applications, Lecture Notes in Comput. Sci. 2542, Springer, Berlin, 2003, pp. 158–166.

[4] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[5] M. Benzi and V. Simoncini, *On the eigenvalues of a class of saddle point matrices*, Numer. Math., 103 (2006), pp. 173–196.

[6] J. H. Bramble and J. E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., 50 (1988), pp. 1–17.

[7] C. Carstensen, M. Kuhn, and U. Langer, *Fast parallel solvers for symmetric boundary element domain decomposition equations*, Numer. Math., 79 (1998), pp. 321–347.

[8] C. R. Dohrmann and R. B. Lehoucq, *A primal-based penalty preconditioner for elliptic saddle point systems*, SIAM J. Numer. Anal., 44 (2006), pp. 270–282.

[9] H. C. Elman, *Multigrid and Krylov subspace methods for the discrete Stokes equations*, in Proceedings of the Seventh Copper Mountain Conference on Multigrid Methods, N. D. Melson, T. A. Manteuffel, S. F. McCormick, and C. C. Douglas, eds., Vol. CP 3339, Hampton, VA, 1996, NASA, Washington, DC, pp. 283–299.

[10] H. C. Elman, A. Ramage, and D. J. Silvester, *Algorithm* 866*: IFISS, a* Matlab *toolbox for modelling incompressible flow*, ACM Trans. Math. Softw., 33 (2007), pp. 1–18.

[11] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.

[12] V. Faber, J. Liesen, and P. Tichý, *The Faber–Manteuffel theorem for linear operators*, SIAM J. Numer. Anal., 46 (2008), pp. 1323–1337.

[13] V. Faber and T. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.
[14] B. Fischer, *Polynomial Based Iteration Methods for Symmetric Linear Systems*, Wiley-Teubner Series Advances in Numerical Mathematics, John Wiley & Sons Ltd., Chichester, 1996.
[15] B. Fischer, A. Ramage, D. J. Silvester, and A. J. Wathen, *Minimum residual methods for augmented systems*, BIT, 38 (1998), pp. 527–543.
[16] R. Freund, *Transpose-free quasi-minimal residual methods for non-Hermitian linear systems*, Numerical analysis manuscript 92-97, AT&T Bell Labs, 14, 1992, pp. 470–482.
[17] R. W. Freund, *Transpose-free quasi-minimal residual methods for non-Hermitian linear systems*, in Recent Advances in Iterative Methods, IMA Vol. Math. Appl. 60, Springer, New York, 1994, pp. 69–94.
[18] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput., 14 (1993), pp. 137–158.
[19] R. W. Freund and N. M. Nachtigal, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.
[20] R. W. Freund and N. M. Nachtigal, *An implementation of the QMR method based on coupled two-term recurrences*, SIAM J. Sci. Comput., 15 (1994), pp. 313–337.
[21] R. W. Freund and N. M. Nachtigal, *Software for simplified Lanczos and QMR algorithms*, Appl. Numer. Math., 19 (1995), pp. 319–341.
[22] G. N. Gatica and N. Heuer, *Conjugate gradient method for dual-dual mixed formulations*, Math. Comp., 71 (2002), pp. 1455–1472.
[23] I. Gohberg, P. Lancaster, and L. Rodman, *Indefinite Linear Algebra and Applications*, Birkhäuser Verlag, Basel, 2005.
[24] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 1996.
[25] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, in Frontiers in Applied Mathematics 17, SIAM, Philadelphia, 1997.
[26] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.
[27] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1990.
[28] A. Klawonn, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), pp. 172–184.
[29] U. Langer and M. Kuhn, *Parallel Solvers for Large-Scale, Coupled Finite and Boundary Element Equations*, unpublished.
[30] J. Liesen, *A Note on the Eigenvalues of Saddle Point Matrices*, Technical report 10-2006, TU Berlin, Berlin, Germany, 2006.
[31] J. Liesen and B. N. Parlett, *On nonsymmetric saddle point matrices that allow conjugate gradient iterations*, Numer. Math., to appear.
[32] J. Liesen and Z. Strakoš, *On optimal short-term recurrences for generating orthogonal Krylov subspace bases*, SIAM Rev., to appear.
[33] G. Meurant, *Computer solution of large linear systems*, in Studies in Mathematics and its Applications 28, North–Holland, Amsterdam, 1999.
[34] A. Meyer and T. Steidten, *Improvements and Experiments on the Bramble–Pasciak Type CG for Mixed Problems in Elasticity*, Technical report, TU Chemnitz, Chemnitz, Germany, 2001.
[35] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[36] B. N. Parlett, D. R. Taylor, and Z. S. Liu, *The look ahead Lánczos algorithm for large unsymmetric eigenproblems*, in Computing Methods in Applied Sciences and Engineering, VI (Versailles, 1983), North–Holland, Amsterdam, 1984, pp. 87–96.
[37] J. Peters, V. Reichelt, and A. Reusken, *Fast iterative solvers for discrete Stokes equations*, SIAM J. Sci. Comput., 27 (2005), pp. 646–666.
[38] Y. Saad, *Iterative methods for sparse linear systems*, 2nd ed., SIAM, Philadelphia, 2003.
[39] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[40] J. Schöberl and W. Zulehner, *Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 752–773.
[41] D. Silvester and A. Wathen, *Fast iterative solution of stabilised Stokes systems.* II. *Using*

general block preconditioners, SIAM J. Numer. Anal., 31 (1994), pp. 1352–1367.
[42] V. Simoncini, *Block triangular preconditioners for symmetric saddle-point problems*, Appl. Numer. Math., 49 (2004), pp. 63–80.
[43] V. Simoncini and D. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–61.
[44] G. L. G. Sleijpen and D. R. Fokkema, *BiCGstab(l) for linear equations involving unsymmetric matrices with complex spectrum*, Electron. Trans. Numer. Anal., 1 (1993), pp. 11–32 (electronic only).
[45] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of biCG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.
[46] H. A. van der Vorst and C. Vuik, *GMRESR: A family of nested GMRES methods*, Numer. Linear Algebra Appl., 1 (1994), pp. 369–386.
[47] A. Wathen and D. Silvester, *Fast iterative solution of stabilised Stokes systems.* I. *Using simple diagonal preconditioners*, SIAM J. Numer. Anal., 30 (1993), pp. 630–649.
[48] W. Zulehner, *Analysis of iterative methods for saddle point problems: A unified approach*, Math. Comp., 71 (2002), pp. 479–505.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 609–638

$\mathcal{H}_2$ MODEL REDUCTION FOR LARGE-SCALE LINEAR DYNAMICAL SYSTEMS*

S. GUGERCIN†, A. C. ANTOULAS‡, AND C. BEATTIE†

Abstract. The optimal $\mathcal{H}_2$ model reduction problem is of great importance in the area of dynamical systems and simulation. In the literature, two independent frameworks have evolved focusing either on solution of Lyapunov equations on the one hand or interpolation of transfer functions on the other, without any apparent connection between the two approaches. In this paper, we develop a new unifying framework for the optimal $\mathcal{H}_2$ approximation problem using best approximation properties in the underlying Hilbert space. This new framework leads to a new set of local optimality conditions taking the form of a structured orthogonality condition. We show that the existing Lyapunov- and interpolation-based conditions are each equivalent to our conditions and so are equivalent to each other. Also, we provide a new elementary proof of the interpolation-based condition that clarifies the importance of the mirror images of the reduced system poles. Based on the interpolation framework, we describe an iteratively corrected rational Krylov algorithm for $\mathcal{H}_2$ model reduction. The formulation is based on finding a reduced order model that satisfies interpolation-based first-order necessary conditions for $\mathcal{H}_2$ optimality and results in a method that is numerically effective and suited for large-scale problems. We illustrate the performance of the method with a variety of numerical experiments and comparisons with existing methods.

Key words. model reduction, rational Krylov, $\mathcal{H}_2$ approximation

AMS subject classifications. 34C20, 41A05, 49K15, 49M05, 93A15, 93C05, 93C15

DOI. 10.1137/060666123

1. Introduction.

Given a dynamical system described by a set of first-order differential equations, the model reduction problem seeks to replace this original set of equations with a (much) smaller set of such equations so that the behavior of both systems is similar in an appropriately defined sense. Such situations arise frequently when physical systems need to be simulated or controlled; the greater the level of detail that is required, the greater the number of resulting equations. In large-scale settings, computations become infeasible due to limitations on computational resources as well as growing inaccuracy due to numerical ill-conditioning. In all these cases the number of equations involved may range from a few hundred to a few million. Examples of large-scale systems abound, ranging from the design of VLSI (very large scale integration) chips to the simulation and control of MEMS (microelectromechanical system) devices. For an overview of model reduction for large-scale dynamical systems we refer to the book [2]. See also [23] for a recent collection of large-scale benchmark problems.

In this paper, we consider single input/single output (SISO) linear dynamical systems represented as

$$G : \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}\,u(t) \\ y(t) = \mathbf{c}^T\mathbf{x}(t) \end{cases} \qquad \text{or} \qquad G(s) = \mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}, \tag{1.1}$$

*Received by the editors July 26, 2006; accepted for publication (in revised form) by P. Benner February 25, 2008; published electronically June 6, 2008.
http://www.siam.org/journals/simax/30-2/66612.html

†Department of Mathematics, Virginia Tech, Blacksburg, VA (gugercin@vt.edu, beattie@vt.edu). The work of these authors was supported in part by the NSF through grants DMS-050597 and DMS-0513542, and by the AFOSR through grant FA9550-05-1-0449.

‡Department of Electrical and Computer Engineering, Rice University, Houston, TX (aca@ece.rice.edu). The work of this author was supported in part by the NSF through grants CCR-0306503 and ACI-0325081.

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$; we define $\mathbf{x}(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}$, $y(t) \in \mathbb{R}$ as the , , , and , respectively, of the system. (We comment on extensions to the multiple input/multiple output (MIMO) case in section 3.2.1, but will confine our analysis and examples to the SISO case.)

$G(s)$ is the transfer function of the system: if $\widehat{u}(s)$ and $\widehat{y}(s)$ denote the Laplace transforms of the input and output $u(t)$ and $y(t)$, respectively, then $\widehat{y}(s) = G(s)\widehat{u}(s)$. With a standard abuse of notation, we will denote both the system and its transfer function by G. The "dimension of G" is taken to be the dimension of the underlying state space, $\dim G = n$ in this case. It will always be assumed that the system, G, is , that is, that the eigenvalues of $\mathbf{A}$ have strictly negative real parts.

The model reduction process will yield another system,

$$G_r : \begin{cases} \dot{\mathbf{x}}_r(t) = \mathbf{A}_r\mathbf{x}_r(t) + \mathbf{b}_r u(t) \\ y_r(t) = \mathbf{c}_r^T\mathbf{x}_r(t) \end{cases} \quad \text{or} \quad G_r(s) = \mathbf{c}_r^T(s\mathbf{I} - \mathbf{A}_r)^{-1}\mathbf{b}_r, \tag{1.2}$$

having (much) smaller dimension $r \ll n$, with $\mathbf{A}_r \in \mathbb{R}^{r\times r}$ and $\mathbf{b}_r$, $\mathbf{c}_r \in \mathbb{R}^r$.

We want $y_r(t) \approx y(t)$ over a large class of inputs $u(t)$. Different measures of approximation and different choices of input classes will lead to different model reduction goals. Suppose one wants to ensure that $\max_{t>0} |y(t) - y_r(t)|$ is small uniformly over all inputs, $u(t)$, having bounded "energy," that is, $\int_0^\infty |u(t)|^2\, dt \le 1$. Observe first that $\widehat{y}(s) - \widehat{y}_r(s) = [G(s) - G_r(s)]\, \widehat{u}(s)$ and then

$$\begin{aligned}
\max_{t>0} |y(t) - y_r(t)| &= \max_{t>0} \left| \frac{1}{2\pi} \int_{-\infty}^{\infty} (\widehat{y}(\imath\omega) - \widehat{y}_r(\imath\omega))\, e^{\imath\omega t}\, d\omega \right| \\
&\le \frac{1}{2\pi} \int_{-\infty}^{\infty} |\widehat{y}(\imath\omega) - \widehat{y}_r(\imath\omega)|\, d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(\imath\omega) - G_r(\imath\omega)|\, |\widehat{u}(\imath\omega)|\, d\omega \\
&\le \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |G(\imath\omega) - G_r(\imath\omega)|^2\, d\omega \right)^{1/2} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |\widehat{u}(\imath\omega)|^2\, d\omega \right)^{1/2} \\
&\le \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |G(\imath\omega) - G_r(\imath\omega)|^2\, d\omega \right)^{1/2} \left(\int_0^{\infty} |u(t)|^2\, dt \right)^{1/2} \\
&\le \left(\frac{1}{2\pi} \int_{-\infty}^{+\infty} |G(\imath\omega) - G_r(\imath\omega)|^2\, d\omega \right)^{1/2} \stackrel{\text{def}}{=} \|G - G_r\|_{\mathcal{H}_2}.
\end{aligned}$$

We seek a reduced order dynamical system, G_r, such that

(i) $\|G - G_r\|_{\mathcal{H}_2}$, the "$\mathcal{H}_2$ error," is as small as possible;
(ii) critical system properties for G (such as stability) exist also in G_r; and
(iii) the computation of G_r (i.e., the computation of $\mathbf{A}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r$) is both efficient and numerically stable.

The problem of finding reduced order models that yield a small $\mathcal{H}_2$ error has been the object of many investigations; see, for instance, [6, 37, 34, 9, 21, 26, 22, 36, 25, 13] and the references therein. Finding a global minimizer of $\|G - G_r\|_{\mathcal{H}_2}$ is a hard task, so the goal in making $\|G - G_r\|_{\mathcal{H}_2}$ "as small as possible" becomes, as for many optimization problems, identification of reduced order models, G_r, that satisfy first-order necessary conditions for local optimality. There is a wide variety of such conditions that may be derived, yet their interconnections are generally unclear. Most methods that can identify reduced order models satisfying such first-order necessary conditions will require dense matrix operations, typically the solution of a sequence of matrix Lyapunov equations, a task which becomes computationally intractable

rapidly as the dimension increases. Such methods are unsuitable even for medium-scale problems. In section 2, we review the moment matching problem for model reduction, its connection with rational Krylov methods (which are very useful for large-scale problems), and basic features of the $\mathcal{H}_2$ norm and inner product.

We offer in section 3 what appears to be a new set of first-order necessary conditions for local optimality of a reduced order model comprising in effect a structured orthogonality condition. We also show its equivalence with two other $\mathcal{H}_2$ optimality conditions that have been previously known (thus showing them all to be equivalent).

An iterative algorithm that is designed to force optimality with respect to a set of conditions that is computationally tractable is described in section 4. The proposed method also forces optimality with respect to the other equivalent conditions as well. It is based on computationally effective use of rational Krylov subspaces and so is suitable for systems whose dimension n is of the order of many thousands of state variables. Numerical examples are presented in section 5.

2. Background.

2.1. Model reduction by moment matching. Given the system (1.1), reduction by [illegible] consists in finding a system (1.2) so that $G_r(s)$ interpolates the values of $G(s)$, and perhaps also derivative values as well, at selected interpolation points (also called [illegible]) σ_k in the complex plane. For our purposes, simple Hermite interpolation suffices, so our problem is to find $\mathbf{A}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r$ so that

$$G_r(\sigma_k) = G(\sigma_k) \quad \text{and} \quad G_r'(\sigma_k) = G'(\sigma_k) \quad \text{for} \quad k = 1, \ldots, r$$

or, equivalently,

$$\mathbf{c}^T(\sigma_k\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} = \mathbf{c}_r^T(\sigma_k\mathbf{I} - \mathbf{A}_r)^{-1}\mathbf{b}_r \quad \text{and} \quad \mathbf{c}^T(\sigma_k\mathbf{I} - \mathbf{A})^{-2}\mathbf{b} = \mathbf{c}_r^T(\sigma_k\mathbf{I} - \mathbf{A}_r)^{-2}\mathbf{b}_r$$

for $k = 1, \ldots, r$. The quantity $\mathbf{c}^T(\sigma_k\mathbf{I} - \mathbf{A})^{-(j+1)}\mathbf{b}$ is called the jth moment of $G(s)$ at σ_k. Moment matching for finite $\sigma \in \mathbb{C}$ becomes [illegible]; see, for example, [3]. Importantly, these problems can be solved in a recursive and numerically effective way by means of [illegible] procedures.

To see this we first consider reduced order models that are constructed by Galerkin approximation: Let $\mathcal{V}_r$ and $\mathcal{W}_r$ be given r-dimensional subspaces of $\mathbb{R}^n$ that are generic in the sense that $\mathcal{V}_r \cap \mathcal{W}_r^\perp = \{0\}$. Then for any input $u(t)$ the reduced order output $y_r(t)$ is defined by

$$\begin{aligned} &\text{Find} \quad \mathbf{v}(t) \in \mathcal{V}_r \quad \text{such that} \quad \dot{\mathbf{v}}(t) - \mathbf{A}\mathbf{v}(t) - \mathbf{b}u(t) \perp \mathcal{W}_r \quad \text{for all } t; \\ &\qquad\qquad \text{then} \quad y_r(t) \stackrel{\text{def}}{=} \mathbf{c}^T\mathbf{v}(t). \end{aligned} \tag{2.1}$$

Denote by $\mathrm{Ran}(\mathbf{M})$ the range of a matrix $\mathbf{M}$. Let $\mathbf{V}_r \in \mathbb{R}^{n\times r}$ and $\mathbf{W}_r \in \mathbb{R}^{n\times r}$ be matrices defined so that $\mathcal{V}_r = \mathrm{Ran}(\mathbf{V}_r)$ and $\mathcal{W}_r = \mathrm{Ran}(\mathbf{W}_r)$. Then the assumption $\mathcal{V}_r \cap \mathcal{W}_r^\perp = \{0\}$ is equivalent to $\mathbf{W}_r^T\mathbf{V}_r$ being nonsingular. The Galerkin approximation (2.1) can be interpreted as $\mathbf{v}(t) = \mathbf{V}_r\mathbf{x}_r(t)$ with $\mathbf{x}_r(t) \in \mathbb{R}^r$ for each t and

$$\mathbf{W}_r^T\left(\mathbf{V}_r\dot{\mathbf{x}}_r(t) - \mathbf{A}\mathbf{V}_r\mathbf{x}_r(t) - \mathbf{b}\,u(t)\right) = 0$$

leading then to the reduced order model (1.2) with

$$\mathbf{A}_r = (\mathbf{W}_r^T\mathbf{V}_r)^{-1}\mathbf{W}_r^T\mathbf{A}\mathbf{V}_r, \quad \mathbf{b}_r = (\mathbf{W}_r^T\mathbf{V}_r)^{-1}\mathbf{W}_r^T\mathbf{b}, \quad \text{and} \quad \mathbf{c}_r^T = \mathbf{c}^T\mathbf{V}_r. \tag{2.2}$$

Evidently the choice of $\mathbf{V}_r$ and $\mathbf{W}_r$ determines the quality of the reduced order model.

Rational interpolation by projection was first proposed by Skelton et al. in [11, 38, 39]. Grimme [17] showed how one can obtain the required projection using the rational Krylov method of Ruhe [33]. Krylov-based methods are able to match moments without ever computing them explicitly. This is important since the computation of moments is in general ill-conditioned. This is a fundamental motivation behind the Krylov-based methods [12].

In Lemma 2.1 and Corollary 2.2 below, we present new short proofs of rational interpolation by Krylov projection that are substantially simpler than those found in the original works [17, 11, 38, 39].

LEMMA 2.1. $\sigma \in \mathbb{C}$ $\mathbf{A}$ $\mathbf{A}_r$

(2.3) $(\sigma\mathbf{I}-\mathbf{A})^{-1}\mathbf{b} \in \mathcal{V}_r$, $G_r(\sigma) = G(\sigma)$.

(2.4) $(\overline{\sigma}\,\mathbf{I}-\mathbf{A}^T)^{-1}\mathbf{c} \in \mathcal{W}_r$, $G_r(\sigma) = G(\sigma)$.

$(\sigma\mathbf{I}-\mathbf{A})^{-1}\mathbf{b} \in \mathcal{V}_r$ $(\overline{\sigma}\,\mathbf{I}-\mathbf{A}^T)^{-1}\mathbf{c} \in \mathcal{W}_r$,

(2.5) $G_r(\sigma) = G(\sigma)$ $G_r'(\sigma) = G'(\sigma)$.

Define $\mathbf{N}_r(z) = \mathbf{V}_r(z\mathbf{I}-\mathbf{A}_r)^{-1}(\mathbf{W}_r^T\mathbf{V}_r)^{-1}\mathbf{W}_r^T(z\mathbf{I}-\mathbf{A})$ and $\widetilde{\mathbf{N}}_r(z) = (z\mathbf{I}-\mathbf{A})\mathbf{N}_r(z)(z\mathbf{I}-\mathbf{A})^{-1}$. Both $\mathbf{N}_r(z)$ and $\widetilde{\mathbf{N}}_r(z)$ are analytic matrix-valued functions in a neighborhood of $z=\sigma$. One may directly verify that $\mathbf{N}_r^2(z) = \mathbf{N}_r(z)$ and $\widetilde{\mathbf{N}}_r^2(z) = \widetilde{\mathbf{N}}_r(z)$ and that $\mathcal{V}_r = \operatorname{Ran}\mathbf{N}_r(z) = \operatorname{Ker}(\mathbf{I}-\mathbf{N}_r(z))$ and $\mathcal{W}_r^{\perp} = \operatorname{Ker}\widetilde{\mathbf{N}}_r(z) = \operatorname{Ran}\big(\mathbf{I}-\widetilde{\mathbf{N}}_r(z)\big)$ for all z in a neighborhood of σ. Then

$$G(z)-G_r(z) = \big[(z\mathbf{I}-\mathbf{A}^T)^{-1}\mathbf{c}\big]^T\Big(\mathbf{I}-\widetilde{\mathbf{N}}_r(z)\Big)(z\mathbf{I}-\mathbf{A})\Big(\mathbf{I}-\mathbf{N}_r(z)\Big)(z\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}.$$

Evaluating at $z=\sigma$ leads to (2.3) and (2.4). Evaluating at $z=\sigma+\varepsilon$ and observing that $(\sigma\mathbf{I}+\varepsilon\mathbf{I}-\mathbf{A})^{-1} = (\sigma\mathbf{I}-\mathbf{A})^{-1} - \varepsilon(\sigma\mathbf{I}-\mathbf{A})^{-2} + \mathcal{O}(\varepsilon^2)$ yields

$$G(\sigma+\varepsilon)-G_r(\sigma+\varepsilon) = \mathcal{O}(\varepsilon^2),$$

which gives (2.5) as a consequence. ∎

COROLLARY 2.2. G $\mathbf{A},\mathbf{b},\mathbf{c}$ $\{\sigma_k\}_{k=1}^r$ $\mathbf{V}_r$ $\mathbf{W}_r$

(2.6) $\operatorname{Ran}(\mathbf{V}_r) = \operatorname{span}\big\{(\sigma_1\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}, \quad \ldots, \quad (\sigma_r\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}\big\}$

(2.7) $\operatorname{Ran}(\mathbf{W}_r) = \operatorname{span}\big\{(\sigma_1\mathbf{I}-\mathbf{A}^T)^{-1}\mathbf{c}, \quad \ldots, \quad (\sigma_r\mathbf{I}-\mathbf{A}^T)^{-1}\mathbf{c}\big\}.$

$\mathbf{V}_r$ $\mathbf{W}_r$ G_r $\mathbf{A}_r = (\mathbf{W}_r^T\mathbf{V}_r)^{-1}\mathbf{W}_r^T\mathbf{A}\mathbf{V}_r$, $\mathbf{b}_r = (\mathbf{W}_r^T\mathbf{V}_r)^{-1}\mathbf{W}_r^T\mathbf{b}$, $\mathbf{c}_r^T = \mathbf{c}^T\mathbf{V}_r$ $G(s)$ σ_k $G(\sigma_k) = G_r(\sigma_k)$ $G'(\sigma_k) = G_r'(\sigma_k)$, $k=1,\ldots,r$

For Krylov-based model reduction, one chooses interpolation points and then constructs $\mathbf{V}_r$ and $\mathbf{W}_r$ satisfying (2.6) and (2.7), respectively. Note that, in a numerical implementation, one does not actually compute $(\sigma_i\mathbf{I}-\mathbf{A})^{-1}$, but instead computes a (potentially sparse) factorization (one for each interpolation point σ_i), uses it to solve a system of equations having $\mathbf{b}$ as a right-hand side, and uses its transpose to solve a system of equations having $\mathbf{c}$ as a right-hand side. The interpolation points are chosen so as to minimize the deviation of G_r from G in a sense that is detailed in the next section. Unlike Gramian-based model reduction methods such as balanced truncation (see section 2.2 below and [2]), Krylov-based model reduction requires only

matrix-vector multiplications and some sparse linear solvers, and can be iteratively implemented; hence it is computationally effective; for details, see also [15, 16].

2.2. Model reduction by balanced truncation. One of the most common model reduction techniques is [illegible] [28, 27]. In this case, the modeling subspaces $\mathbf{V}_r$ and $\mathbf{W}_r$ depend on the solutions to the two Lyapunov equations

$$\mathbf{AP} + \mathbf{PA}^T + \mathbf{bb}^T = \mathbf{0}, \qquad \mathbf{A}^T\mathbf{Q} + \mathbf{QA} + \mathbf{c}^T\mathbf{c} = \mathbf{0}. \tag{2.8}$$

$\mathbf{P}$ and $\mathbf{Q}$ are called the reachability and observability Gramians, respectively. Under the assumption that $\mathbf{A}$ is stable, both $\mathbf{P}$ and $\mathbf{Q}$ are positive semidefinite matrices. Square roots of the eigenvalues of the product $\mathbf{PQ}$ are the singular values of the Hankel operator associated with $G(s)$ and are called the Hankel singular values of $G(s)$, denoted by $\eta_i(G)$.

Let $\mathbf{P} = \mathbf{UU}^T$ and $\mathbf{Q} = \mathbf{LL}^T$. Let $\mathbf{U}^T\mathbf{L} = \mathbf{ZSY}^T$ be the singular value decomposition with $\mathbf{S} = \operatorname{diag}(\eta_1, \eta_2, \ldots, \eta_n)$. Let $\mathbf{S}_r = \operatorname{diag}(\eta_1, \eta_2, \ldots, \eta_r)$, $r < n$. Construct

$$\mathbf{W}_r = \mathbf{LY}_r\mathbf{S}_r^{-1/2} \qquad \text{and} \qquad \mathbf{V}_r = \mathbf{UZ}_r\mathbf{S}_r^{-1/2}, \tag{2.9}$$

where $\mathbf{Z}_r$ and $\mathbf{Y}_r$ denote the leading r columns of left singular vectors, $\mathbf{Z}$, and right singular vectors, $\mathbf{Y}$, respectively. The rth-order reduced order model via [illegible], $G_r(s)$, is obtained by reducing $G(s)$ using $\mathbf{W}_r$ and $\mathbf{V}_r$ from (2.9).

Another important dynamical systems norm (besides the $\mathcal{H}_2$ norm) is the $\mathcal{H}_\infty$ norm defined as $\|G\|_{\mathcal{H}_\infty} := \sup_{\omega\in\mathbb{R}} |G(\imath\omega)|$. The reduced order system $G_r(s)$ obtained by balanced truncation is asymptotically stable and the $\mathcal{H}_\infty$ norm of the error system satisfies $\|G - G_r\|_{\mathcal{H}_\infty} \le 2(\eta_{r+1} + \cdots + \eta_n)$.

The value of having, for reduced order models, guaranteed stability and an explicit error bound is widely recognized, though it is achieved at potentially considerable cost. As described above, balanced truncation requires the solution of two Lyapunov equations of order n, which is a formidable task in large-scale settings. For more details and background on balanced truncation, see section III.7 of [2].

2.3. The $\mathcal{H}_2$ norm. $\mathcal{H}_2$ will denote the set of functions, $g(z)$, that are analytic for z in the open right half plane, $\operatorname{Re}(z) > 0$, and such that for each fixed $\operatorname{Re}(z) = x > 0$, $g(x + \imath y)$ is square integrable as a function of $y \in (-\infty, \infty)$ in such a way that

$$\sup_{x>0} \int_{-\infty}^{\infty} |g(x + \imath y)|^2 \, dy < \infty.$$

$\mathcal{H}_2$ is a Hilbert space and holds our interest because transfer functions associated with stable SISO finite-dimensional dynamical systems are elements of $\mathcal{H}_2$. Indeed, if $G(s)$ and $H(s)$ are transfer functions associated with real stable SISO dynamical systems, then the $\mathcal{H}_2$ inner product can be defined as

$$\langle G,\ H\rangle_{\mathcal{H}_2} \stackrel{\text{def}}{=} \frac{1}{2\pi}\int_{-\infty}^{\infty} \overline{G(\imath\omega)}H(\imath\omega)\, d\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} G(-\imath\omega)H(\imath\omega)\, d\omega, \tag{2.10}$$

with a norm defined as

$$\|G\|_{\mathcal{H}_2} \stackrel{\text{def}}{=} \left(\frac{1}{2\pi}\int_{-\infty}^{+\infty} |G(\imath\omega)|^2\, d\omega\right)^{1/2}. \tag{2.11}$$

Notice in particular that if $G(s)$ and $H(s)$ represent real dynamical systems, then $\langle G,\ H\rangle_{\mathcal{H}_2} = \langle H,\ G\rangle_{\mathcal{H}_2}$ and $\langle G,\ H\rangle_{\mathcal{H}_2}$ must be real.

There are two alternate characterizations of this inner product that make it far more computationally accessible.

LEMMA 2.3. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{m\times m}$, $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$, $\widetilde{\mathbf{b}}, \widetilde{\mathbf{c}} \in \mathbb{R}^m$, be such that

$$G(s) = \mathbf{c}^T (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{b} \quad \text{and} \quad H(s) = \widetilde{\mathbf{c}}^T (s\mathbf{I} - \mathbf{B})^{-1} \widetilde{\mathbf{b}}.$$

Then $\langle G,\ H\rangle_{\mathcal{H}_2}$ can be computed in the following ways:

(2.12) $\mathbf{P}$ satisfies $\mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{B}^T + \mathbf{b}\widetilde{\mathbf{b}}^T = 0$, then $\langle G,\ H\rangle_{\mathcal{H}_2} = \mathbf{c}^T\mathbf{P}\widetilde{\mathbf{c}}$.

(2.13) $\mathbf{Q}$ satisfies $\mathbf{Q}\mathbf{A} + \mathbf{B}^T\mathbf{Q} + \widetilde{\mathbf{c}}\mathbf{c}^T = 0$, then $\langle G,\ H\rangle_{\mathcal{H}_2} = \widetilde{\mathbf{b}}^T\mathbf{Q}\mathbf{b}$.

(2.14) $\mathbf{R}$ satisfies $\mathbf{A}\mathbf{R} + \mathbf{R}\mathbf{B} + \mathbf{b}\widetilde{\mathbf{c}}^T = 0$, then $\langle G,\ H\rangle_{\mathcal{H}_2} = \mathbf{c}^T\mathbf{R}\widetilde{\mathbf{b}}$.

In the case where $\mathbf{A} = \mathbf{B}$, $\mathbf{b} = \widetilde{\mathbf{b}}$, and $\mathbf{c} = \widetilde{\mathbf{c}}$, the matrix $\mathbf{P}$ is the "observability gramian" of $G(s)$, $\mathbf{Q}$ is the "controllability gramian" of $G(s)$, and $\mathbf{R}$ is the "cross gramian" of $G(s)$:

$$\|G\|_{\mathcal{H}_2}^2 = \mathbf{c}^T\mathbf{P}\mathbf{c} = \mathbf{b}^T\mathbf{Q}\mathbf{b} = \mathbf{c}^T\mathbf{R}\mathbf{b}. \tag{2.15}$$

Gramians play a prominent role in the analysis of linear dynamical systems; refer to [2] for more information.

We detail the proof of (2.12); proofs of (2.13) and (2.14) are similar. Since $\mathbf{A}$ and $\mathbf{B}$ are stable, the solution, $\mathbf{P}$, to the Sylvester equation of (2.12) exists and is unique. For any $\omega \in \mathbb{R}$, rearrange this equation to obtain in sequence

$$\begin{aligned}
(-\imath\omega\mathbf{I} - \mathbf{A})\,\mathbf{P} + \mathbf{P}\left(\imath\omega\mathbf{I} - \mathbf{B}^T\right) - \mathbf{b}\widetilde{\mathbf{b}}^T &= 0,\\
(-\imath\omega\mathbf{I} - \mathbf{A})^{-1}\,\mathbf{P} + \mathbf{P}\left(\imath\omega\mathbf{I} - \mathbf{B}^T\right)^{-1} &= (-\imath\omega\mathbf{I} - \mathbf{A})^{-1}\,\mathbf{b}\widetilde{\mathbf{b}}^T\left(\imath\omega\mathbf{I} - \mathbf{B}^T\right)^{-1},\\
\mathbf{c}^T(-\imath\omega\mathbf{I} - \mathbf{A})^{-1}\,\mathbf{P}\widetilde{\mathbf{c}} + \mathbf{c}^T\mathbf{P}\left(\imath\omega\mathbf{I} - \mathbf{B}^T\right)^{-1}\widetilde{\mathbf{c}} &= G(-\imath\omega)H(\imath\omega),
\end{aligned}$$

and finally

$$\begin{aligned}
\mathbf{c}^T\left(\int_{-L}^{L}(-\imath\omega\mathbf{I} - \mathbf{A})^{-1}\,d\omega\right)\mathbf{P}\widetilde{\mathbf{c}} + \mathbf{c}^T\mathbf{P}\left(\int_{-L}^{L}\left(\imath\omega\mathbf{I} - \mathbf{B}^T\right)^{-1}\,d\omega\right)\widetilde{\mathbf{c}}&\\
= \int_{-L}^{L} G(-\imath\omega)H(\imath\omega)\,d\omega.&
\end{aligned}$$

Taking $L \to \infty$ and using Lemma A.1 in the appendix leads to

$$\begin{aligned}
\int_{-\infty}^{\infty} G(-\imath\omega)H(\imath\omega)\,d\omega &= \mathbf{c}^T\left(\text{P.V.}\int_{-\infty}^{\infty}(-\imath\omega\mathbf{I} - \mathbf{A})^{-1}\,d\omega\right)\mathbf{P}\widetilde{\mathbf{c}}\\
&\quad + \mathbf{c}^T\mathbf{P}\left(\text{P.V.}\int_{-\infty}^{\infty}\left(\imath\omega\mathbf{I} - \mathbf{B}^T\right)^{-1}\,d\omega\right)\widetilde{\mathbf{c}}\\
&= 2\pi\ \mathbf{c}^T\mathbf{P}\widetilde{\mathbf{c}}. \qquad \square
\end{aligned}$$

Recently, Antoulas [2] obtained a new expression for $\|G\|_{\mathcal{H}_2}$ based on the poles and residues of the transfer function $G(s)$ that complements the widely known alternative expression (2.15). We provide a compact derivation of this expression and the associated $\mathcal{H}_2$ inner product.

If $f(s)$ is a meromorphic function with a pole at λ, denote the residue of $f(s)$ at λ by $\mathsf{res}[f(s), \lambda]$. Thus, if λ is a simple pole of $f(s)$, then $\mathsf{res}[f(s), \lambda] = \lim_{s\to\lambda}(s-\lambda)f(s)$, and if λ is a double pole of $f(s)$, then $\mathsf{res}[f(s), \lambda] = \lim_{s\to\lambda} \frac{d}{ds}\left[(s-\lambda)^2 f(s)\right]$.

LEMMA 2.4. [illegible] $G(s)$ [illegible] $\lambda_1, \lambda_2, \ldots, \lambda_n$ [illegible] $H(s)$ [illegible] $\mu_1, \mu_2, \ldots, \mu_m$ [illegible]

$$\langle G,\ H\rangle_{\mathcal{H}_2} = \sum_{k=1}^{m} \mathsf{res}[G(-s)H(s), \mu_k] = \sum_{k=1}^{n} \mathsf{res}[H(-s)G(s), \lambda_k]. \tag{2.16}$$

[illegible]

- [illegible] μ_k [illegible] $H(s)$ [illegible]
$$\mathsf{res}[G(-s)H(s), \mu_k] = G(-\mu_k)\mathsf{res}[H(s), \mu_k];$$
- [illegible] μ_k [illegible] $H(s)$ [illegible]
$$\mathsf{res}[G(-s)H(s), \mu_k] = G(-\mu_k)\,\mathsf{res}[H(s), \mu_k] - G'(-\mu_k)\cdot h_0(\mu_k),$$
[illegible] $h_0(\mu_k) = \lim_{s\to\mu_k}\left((s-\mu_k)^2 H(s)\right)$.

[illegible] Notice that the function $G(-s)H(s)$ has singularities at $\mu_1, \mu_2, \ldots, \mu_m$ and $-\lambda_1, -\lambda_2, \ldots, -\lambda_n$. For any $R > 0$, define the semicircular contour in the left half plane:

$$\Gamma_R = \{z\,|z = \imath\omega \text{ with } \omega \in [-R, R]\,\} \cup \left\{ z \,\middle|\, z = R\,e^{\imath\theta} \text{ with } \theta \in \left[\frac{\pi}{2}, \frac{3\pi}{2}\right] \right\}.$$

Γ_R bounds a region that for sufficiently large R contains all the system poles of $H(s)$ and so, by the residue theorem,

$$\begin{aligned}\langle G,\ H\rangle_{\mathcal{H}_2} &= \frac{1}{2\pi}\int_{-\infty}^{+\infty} G(-\imath\omega)H(\imath\omega)\,d\omega \\ &= \lim_{R\to\infty}\frac{1}{2\pi\imath}\int_{\Gamma_R} G(-s)H(s)\ ds = \sum_{k=1}^{m}\mathsf{res}[G(-s)H(s), \mu_k].\end{aligned}$$

Evidently, if μ_k is a simple pole for $H(s)$, it is also a simple pole for $G(-s)H(s)$ and

$$\mathsf{res}[G(-s)H(s),\ \mu_k] = \lim_{s\to\mu_k}(s-\mu_k)G(-s)H(s) = G(-\mu_k)\lim_{s\to\mu_k}(s-\mu_k)H(s).$$

If μ_k is a double pole for $H(s)$, then it is also a double pole for $G(-s)H(s)$ and

$$\begin{aligned}\mathsf{res}[G(-s)H(s),\ \mu_k] &= \lim_{s\to\mu_k}\frac{d}{ds}(s-\mu_k)^2 G(-s)H(s) \\ &= \lim_{s\to\mu_k} G(-s)\frac{d}{ds}(s-\mu_k)^2 H(s) - G'(-s)(s-\mu_k)^2 H(s) \\ &= G(-\mu_k)\lim_{s\to\mu_k}\frac{d}{ds}(s-\mu_k)^2 H(s) - G'(-\mu_k)\lim_{s\to\mu_k}(s-\mu_k)^2 H(s). \qquad \square\end{aligned}$$

Lemma 2.4 immediately yields the expression for $\|G\|_{\mathcal{H}_2}$ given by Antoulas [2, p. 145] based on poles and residues of the transfer function $G(s)$.

COROLLARY 2.5. [illegible] $G(s)$ [illegible] $\lambda_1, \lambda_2, \ldots, \lambda_n$, [illegible]

$$\|G\|_{\mathcal{H}_2} = \left(\sum_{k=1}^{n}\mathsf{res}[G(s), \lambda_k]G(-\lambda_k)\right)^{1/2}.$$

3. Optimal $\mathcal{H}_2$ model reduction. In this section, we investigate three frameworks of necessary conditions for $\mathcal{H}_2$ optimality. The first utilizes the inner product structure $\mathcal{H}_2$ and leads to what could be thought of as a geometric condition for optimality. This appears to be a new characterization of $\mathcal{H}_2$ optimality for reduced order models. The remaining two frameworks, interpolation-based [26] and Lyapunov-based [36, 22], are easily derived from the first framework and in this way can be seen to be equivalent to one another—a fact that is not a priori evident. This equivalence proves that solving the optimal $\mathcal{H}_2$ problem in the Krylov framework is equivalent to solving it in the Lyapunov framework, which leads to the proposed Krylov-based method for $\mathcal{H}_2$ model reduction in section 4.

Given G, a stable SISO finite-dimensional dynamical system as described in (1.1), we seek a stable reduced order system G_r of order r as described in (1.2), which is the best stable rth-order dynamical system approximating G with respect to the $\mathcal{H}_2$ norm:

$$\|G - G_r\|_{\mathcal{H}_2} = \min_{\substack{\dim(\widetilde{G}_r)=r \\ \widetilde{G}_r \text{ : stable}}} \|G - \widetilde{G}_r\|_{\mathcal{H}_2}. \tag{3.1}$$

Many researchers have worked on problem (3.1), the [illegible] $\mathcal{H}_2$ [illegible]. See [37, 34, 9, 21, 26, 22, 36, 25] and the references therein.

3.1. Structured orthogonality optimality conditions. The set of all stable rth-order dynamical systems do not constitute a subspace of $\mathcal{H}_2$, so the best rth-order $\mathcal{H}_2$ approximation is not so easy to characterize, the Hilbert space structure of $\mathcal{H}_2$ notwithstanding. This observation does suggest the following narrower though simpler result.

THEOREM 3.1. [illegible] $\mu_1, \mu_2, \ldots, \mu_r \subset \mathbb{C}$ [illegible] $\mathcal{M}(\boldsymbol{\mu})$ [illegible] $\mu_1, \mu_2, \ldots, \mu_r$ [illegible]

- $H \in \mathcal{M}(\boldsymbol{\mu})$ [illegible] H [illegible] $\dim(H) = r$
- $\mathcal{M}(\boldsymbol{\mu})$ [illegible] $(r-1)$ [illegible] $\mathcal{H}_2$
- $G_r \in \mathcal{M}(\boldsymbol{\mu})$ [illegible]

$$\|G - G_r\|_{\mathcal{H}_2} = \min_{\widetilde{G}_r \in \mathcal{M}(\boldsymbol{\mu})} \|G - \widetilde{G}_r\|_{\mathcal{H}_2} \tag{3.2}$$

[illegible]

$$\langle G - G_r,\, H \rangle_{\mathcal{H}_2} = 0 \qquad \text{[illegible]} \quad H \in \mathcal{M}(\boldsymbol{\mu}). \tag{3.3}$$

[illegible] G_r [illegible] (3.2) [illegible]

[illegible]. The key observation is that $\mathcal{M}(\boldsymbol{\mu})$ is a closed subspace of $\mathcal{H}_2$. Then the equivalence of (3.2) and (3.3) follows from the classic projection theorem in Hilbert space (cf. [32]). □

One consequence of Theorem 3.1 is that if $G_r(s)$ interpolates a real system $G(s)$ at the [illegible] of its own poles (i.e., at the poles of $G_r(s)$ reflected across the imaginary axis), then $G_r(s)$ is guaranteed to be an [illegible] approximation of $G(s)$ relative to the $\mathcal{H}_2$ norm among all reduced order systems having the same reduced system poles $\{\mu_i\}_{i=1}^r$. An analogous result for optimal rational approximants to analytic functions on the unit disk can be found in [14]. The set of stable rth-order dynamical systems is not convex, and so the original problem (3.1) allows for

multiple minimizers. Indeed there may be "local minimizers" that do not solve (3.1). A reduced order system, G_r, is a [illegible] for (3.1) if, for all $\varepsilon > 0$ sufficiently small,

$$\|G - G_r\|_{\mathcal{H}_2} \leq \|G - \widetilde{G}_r^{(\varepsilon)}\|_{\mathcal{H}_2} \tag{3.4}$$

for all stable dynamical systems $\widetilde{G}_r^{(\varepsilon)}$ with $\dim(\widetilde{G}_r^{(\varepsilon)}) = r$ and $\|G_r - \widetilde{G}_r^{(\varepsilon)}\|_{\mathcal{H}_2} \leq C\,\varepsilon$, with C being a constant that may depend on the particular family $\widetilde{G}_r^{(\varepsilon)}$ considered. As a practical matter, the global minimizers that solve (3.1) are difficult to obtain with certainty; current approaches favor seeking reduced order models that satisfy a local (first-order) necessary condition for optimality. Even though such strategies do not guarantee global minimizers, they often produce effective reduced order models nonetheless. In this spirit, we give necessary conditions for optimality for the reduced order system, G_r, that appear as structured orthogonality conditions similar to (3.3).

THEOREM 3.2. [illegible] G_r [illegible] G [illegible] (3.4) [illegible] G_r [illegible]

$$\langle G - G_r,\ G_r \cdot H_1 + H_2 \rangle_{\mathcal{H}_2} = 0 \tag{3.5}$$

[illegible] H_1 [illegible] H_2 [illegible] G_r

($G_r \cdot H_1$ here denotes pointwise multiplication of scalar functions.)

[illegible] Theorem 3.1 implies (3.5) with $H_1 = 0$, so it suffices to show that the hypotheses imply that $\langle G - G_r,\ G_r \cdot H \rangle_{\mathcal{H}_2} = 0$ for all real dynamical systems H having the same poles with the same multiplicities as G_r.

Suppose that $\{\widetilde{G}_r^{(\varepsilon)}\}_{\varepsilon>0}$ is a family of real stable dynamical systems with $\dim(\widetilde{G}_r^{(\varepsilon)}) = r$ and $\|G_r - \widetilde{G}_r^{(\varepsilon)}\|_{\mathcal{H}_2} < C\varepsilon$ for some constant $C > 0$. Then for all $\varepsilon > 0$ sufficiently small,

$$\begin{aligned}
\|G - G_r\|_{\mathcal{H}_2}^2 &\leq \|G - \widetilde{G}_r^{(\varepsilon)}\|_{\mathcal{H}_2}^2 \\
&\leq \|(G - G_r) + (G_r - \widetilde{G}_r^{(\varepsilon)})\|_{\mathcal{H}_2}^2 \\
&\leq \|G - G_r\|_{\mathcal{H}_2}^2 + 2\left\langle G - G_r,\ G_r - \widetilde{G}_r^{(\varepsilon)} \right\rangle_{\mathcal{H}_2} + \|G_r - \widetilde{G}_r^{(\varepsilon)})\|_{\mathcal{H}_2}^2.
\end{aligned}$$

This in turn implies for all $\varepsilon > 0$ sufficiently small that

$$0 \leq 2\left\langle G - G_r,\ G_r - \widetilde{G}_r^{(\varepsilon)} \right\rangle_{\mathcal{H}_2} + \|G_r - \widetilde{G}_r^{(\varepsilon)}\|_{\mathcal{H}_2}^2. \tag{3.6}$$

By considering a few different "directions of approach" of $\widetilde{G}_r^{(\varepsilon)}$ to G_r as $\varepsilon \to 0$, (3.6) will lead to a few different necessary conditions for G_r to be a locally optimal reduced order model. Denote the poles of G_r as $\mu_1, \mu_2, \ldots, \mu_r$ and suppose they are ordered so that the first m_R are real and the next m_C are in the upper half plane. Write $\mu_i = \alpha_i + \imath\beta_i$. Any real rational function having the same poles as $G_r(s)$ can be written as

$$H(s) = \sum_{i=1}^{m_R} \frac{\gamma_i}{s - \mu_i} + \sum_{i=m_R+1}^{m_R+m_C} \frac{\rho_i(s - \alpha_i) + \tau_i}{(s - \alpha_i)^2 + \beta_i^2},$$

with arbitrary real-valued choices for γ_i, ρ_i, and τ_i. Now suppose that μ is a real pole for G_r and that

$$\left\langle G - G_r,\ \frac{G_r(s)}{s - \mu} \right\rangle_{\mathcal{H}_2} \neq 0. \tag{3.7}$$

Write $G_r(s) = \frac{p_{r-1}(s)}{(s-\mu)\,q_{r-1}(s)}$ for real polynomials $p_{r-1},\ q_{r-1} \in \mathcal{P}_{r-1}$ and define

$$\widetilde{G}_r^{(\varepsilon)}(s) = \frac{p_{r-1}(s)}{[s-\mu-(\pm\varepsilon)]\ q_{r-1}(s)},$$

where the sign of $\pm\varepsilon$ is chosen to match that of $\left\langle G - G_r,\ \frac{G_r(s)}{s-\mu}\right\rangle_{\mathcal{H}_2}$. Then we have

$$\widetilde{G}_r^{(\varepsilon)}(s) = G_r(s) \pm \varepsilon \frac{p_{r-1}(s)}{(s-\mu)^2\, q_{r-1}(s)} + \mathcal{O}(\varepsilon^2),$$

which leads to $G_r(s) - \widetilde{G}_r^{(\varepsilon)}(s) = \mp\varepsilon \frac{G_r(s)}{s-\mu} + \mathcal{O}(\varepsilon^2)$ and

$$\left\langle G - G_r,\ G_r - \widetilde{G}_r^{(\varepsilon)}\right\rangle_{\mathcal{H}_2} = -\varepsilon \left| \left\langle G - G_r,\ \frac{G_r(s)}{s-\mu}\right\rangle_{\mathcal{H}_2}\right| + \mathcal{O}(\varepsilon^2). \tag{3.8}$$

Then (3.6) implies that as $\varepsilon \to 0$, $0 < \left|\left\langle G - G_r,\ \frac{G_r(s)}{s-\mu}\right\rangle_{\mathcal{H}_2}\right| \le C\varepsilon$ for some constant C, which then contradicts (3.7).

Now suppose that $\mu = \alpha + \imath\beta$ is a pole for G_r with a nontrivial imaginary part, $\beta \neq 0$, and so is one of a conjugate pair of poles for G_r. Suppose further that

$$\left\langle G - G_r,\ \frac{G_r(s)}{(s-\alpha)^2+\beta^2}\right\rangle_{\mathcal{H}_2} \neq 0 \quad \text{and} \quad \left\langle G - G_r,\ \frac{(s-\alpha)\,G_r(s)}{(s-\alpha)^2+\beta^2}\right\rangle_{\mathcal{H}_2} \neq 0. \tag{3.9}$$

Write $G_r(s) = \frac{p_{r-1}(s)}{[(s-\alpha)^2+\beta^2]\,q_{r-2}(s)}$ for some choice of real polynomials $p_{r-1} \in \mathcal{P}_{r-1}$ and $q_{r-2} \in \mathcal{P}_{r-2}$. Arguments exactly analogous to the previous case lead to the remaining assertions. In particular,

$$\begin{aligned}
\text{to show} \quad & \left\langle G - G_r,\ \frac{G_r(s)}{(s-\alpha)^2+\beta^2}\right\rangle_{\mathcal{H}_2} = 0, \\
& \text{consider} \quad \widetilde{G}_r^{(\varepsilon)}(s) = \frac{p_{r-1}(s)}{[(s-\alpha)^2+\beta^2-(\pm\varepsilon)]\,q_{r-2}(s)}; \\
\text{to show} \quad & \left\langle G - G_r,\ \frac{(s-\alpha)\,G_r(s)}{(s-\alpha)^2+\beta^2}\right\rangle_{\mathcal{H}_2} = 0, \\
& \text{consider} \quad \widetilde{G}_r^{(\varepsilon)}(s) = \frac{p_{r-1}(s)}{[(s-\alpha-(\pm\varepsilon))^2+\beta^2]\,q_{r-2}(s)}.
\end{aligned}$$

The conclusion follows then by observing that if G_r is a locally optimal $\mathcal{H}_2$ reduced order model, then

$$\begin{aligned}
\langle G - G_r,\ G_r \cdot H_1 + H_2\rangle_{\mathcal{H}_2} = & \sum_{i=1}^{m_R} \gamma_i \left\langle G - G_r,\ \frac{G_r(s)}{s-\mu_i}\right\rangle_{\mathcal{H}_2} \\
& + \sum_{i=m_R+1}^{m_R+m_C} \rho_i \left\langle G - G_r,\ \frac{(s-\alpha_i)\,G_r(s)}{(s-\alpha_i)^2+\beta_i^2}\right\rangle_{\mathcal{H}_2} \\
& + \sum_{i=m_R+1}^{m_R+m_C} \tau_i \left\langle G - G_r,\ \frac{G_r(s)}{(s-\alpha_i)^2+\beta_i^2}\right\rangle_{\mathcal{H}_2} \\
& + \langle G - G_r,\ H_2(s)\rangle_{\mathcal{H}_2} = 0.
\end{aligned}$$

Theorem 3.2 describes new necessary conditions for the $\mathcal{H}_2$ approximation problem as structured orthogonality conditions. This new formulation amounts to a unifying framework for the optimal $\mathcal{H}_2$ problem. Indeed, as we show in sections 3.2 and 3.3, two other known optimality frameworks, namely, interpolatory- [26] and Lyapunov-based conditions [36, 22], can be directly obtained from our new conditions by using an appropriate form for the $\mathcal{H}_2$ inner product. The interpolatory framework uses the residue formulation of the $\mathcal{H}_2$ inner product as in (2.16); the Lyapunov framework uses the Sylvester equation formulation of the $\mathcal{H}_2$ norm as in (2.12).

3.2. Interpolation-based optimality conditions. Corollary 2.5 immediately yields an observation regarding the $\mathcal{H}_2$ norm of the error system, which serves as a main motivation for the interpolation framework of the optimal $\mathcal{H}_2$ problem.

PROPOSITION 3.3. [illegible] $G(s)$ [illegible] $G_r(s)$ [illegible] λ_i [illegible] $\widetilde{\lambda}_i$ [illegible] $G(s)$ [illegible] $G_r(s)$ [illegible] $G_r(s)$ [illegible] ϕ_i [illegible] $\widetilde{\phi}_j$ [illegible] $G(s)$ [illegible] $G_r(s)$ [illegible] λ_i [illegible] $\widetilde{\lambda}_i$ [illegible] $\phi_i = \mathsf{res}[G(s), \lambda_i]$, $i = 1, \ldots, n$ [illegible] $\widetilde{\phi}_j = \mathsf{res}[G_r(s), \widetilde{\lambda}_j]$, $j = 1, \ldots, r$. [illegible] $\mathcal{H}_2$ [illegible]

$$\begin{aligned}\|G - G_r\|_{\mathcal{H}_2}^2 &= \sum_{i=1}^{n} \mathsf{res}\left[\left(G(-s) - G_r(-s)\right)\left(G(s) - G_r(s)\right), \lambda_i\right] \\ &\quad + \sum_{j=1}^{r} \mathsf{res}\left[\left(G(-s) - G_r(-s)\right)\left(G(s) - G_r(s)\right), \widetilde{\lambda}_j\right] \\ &= \sum_{i=1}^{n} \phi_i \Big(G(-\lambda_i) - G_r(-\lambda_i)\Big) - \sum_{j=1}^{r} \widetilde{\phi}_j \Big(G(-\widetilde{\lambda}_j) - G_r(-\widetilde{\lambda}_j)\Big). \end{aligned} \tag{3.10}$$

The $\mathcal{H}_2$ error expression (3.10) is valid for any reduced order model regardless of the underlying reduction technique and generalizes a result of [20, 18] to the most general setting.

Proposition 3.3 has the system-theoretic interpretation that the $\mathcal{H}_2$ error is due to mismatch of the transfer functions $G(s)$ and $G_r(s)$ at mirror images of the full-order poles λ_i and reduced order poles $\widetilde{\lambda}_i$. This expression reveals that for good $\mathcal{H}_2$ performance, $G_r(s)$ should approximate $G(s)$ well at $-\lambda_i$ and $-\widetilde{\lambda}_j$. Note that $\widetilde{\lambda}_i$ is not known a priori. Therefore, to minimize the $\mathcal{H}_2$ error, Gugercin and Antoulas [20] proposed choosing $\sigma_i = -\lambda_i(\mathbf{A})$, where $\lambda_i(\mathbf{A})$ are those system poles having big residuals ϕ_i. They have illustrated that this selection of interpolation points works quite well; see [18, 20]. However, as (3.10) illustrates, there is a second part of the $\mathcal{H}_2$ error due to the mismatch at $-\widetilde{\lambda}_j$. Indeed, as we will show below, interpolation at $-\widetilde{\lambda}_i$ is more important for model reduction and is a necessary condition for optimal $\mathcal{H}_2$ model reduction; i.e., $\sigma_i = -\widetilde{\lambda}_i$ is the optimal shift selection.

THEOREM 3.4. [illegible] $G(s) = \mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}$, $G_r(s) = \mathbf{c}_r^T(s\mathbf{I} - \mathbf{A}_r)^{-1}\mathbf{b}_r$ [illegible] r, [illegible] $\mathcal{H}_2$ [illegible] (3.1) [illegible] $G_r(s)$ [illegible] $\widetilde{\lambda}_i$, $i = 1, \ldots, r$ [illegible] $G_r(s)$ [illegible] $G(s)$ [illegible] $-\widetilde{\lambda}_i$, $i = 1, \ldots, r$

$$G_r(-\widetilde{\lambda}_i) = G(-\widetilde{\lambda}_i) \quad \text{and} \quad G_r'(-\widetilde{\lambda}_i) = G'(-\widetilde{\lambda}_i) \quad \text{for} \quad i = 1, \ldots, r. \tag{3.11}$$

[illegible] From (3.5), consider first the case $H_1 = 0$ and H_2 is an arbitrary transfer function with simple poles at $\widetilde{\lambda}_i$, $i = 1, \ldots, r$. Denote $\widetilde{\phi}_i = \mathsf{res}[H_2(s), \widetilde{\lambda}_i]$. Then (2.16)

leads to

$$\langle G - G_r,\ H_2\rangle_{\mathcal{H}_2} = \sum_{i=1}^{r} \mathsf{res}[(G(-s) - G_r(-s))\, H_2(s), \widetilde{\lambda}_i] = \sum_{i=1}^{r} \widetilde{\phi}_i \left(G(-\widetilde{\lambda}_i) - G_r(-\widetilde{\lambda}_i)\right) = 0.$$

Since this is true for arbitrary choices of $\widetilde{\phi}_i$, we have $G(-\widetilde{\lambda}_i) = G_r(-\widetilde{\lambda}_i)$. Now consider the case $H_2 = 0$ and H_1 is an arbitrary transfer function with simple poles at $\widetilde{\lambda}_i$, $i = 1, \ldots, r$. Then $G_r(s)H_1(s)$ has double poles at $\widetilde{\lambda}_i$, $i = 1, \ldots, r$, and since $G(-\widetilde{\lambda}_i) = G_r(-\widetilde{\lambda}_i)$ we have

$$\langle G - G_r,\ G_r \cdot H_1\rangle_{\mathcal{H}_2} = \sum_{i=1}^{r} \mathsf{res}[(G(-s) - G_r(-s))\, G_r(s)H_1(s), \widetilde{\lambda}_i] = -\sum_{i=1}^{r} \widetilde{\phi}_i\, \mathsf{res}[G_r, \widetilde{\lambda}_i] \left(G'(-\widetilde{\lambda}_i) - G_r'(-\widetilde{\lambda}_i)\right) = 0,$$

where we have calculated

$$\lim_{s\to\widetilde{\lambda}_i} \left((s - \widetilde{\lambda}_i)^2 G_r(s) \cdot H_1(s)\right) = \mathsf{res}[H_1(s), \widetilde{\lambda}_i] \cdot \mathsf{res}[G_r(s), \widetilde{\lambda}_i] = \widetilde{\phi}_i\, \mathsf{res}[G_r, \widetilde{\lambda}_i]. \qquad \square$$

We refer to the first-order conditions (3.11) as Meier–Luenberger conditions, recognizing the work of [26], although we have here directly obtained them from the newly derived structured orthogonality conditions (3.5).

In Theorem 3.4, we assume that the reduced order poles (eigenvalues of $\mathbf{A}_r$) are simple; analogous results for the case that G_r has a higher order pole are straightforward and correspond to interpolation conditions of higher derivatives at the mirror images of reduced order poles.

3.2.1. Multiple input/multiple output systems. Many of these considerations extend naturally to the multiple input/multiple output (MIMO) setting:

$$\mathbf{G}: \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \quad \text{or} \quad \mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}, \tag{3.12}$$

where the state vector $\mathbf{x}(t) \in \mathbb{R}^n$ as before, but now the system has an input $\mathbf{u}(t) \in \mathbb{R}^m$ and output $\mathbf{y}(t) \in \mathbb{R}^p$, so that $\mathbf{B} \in \mathbb{R}^{n\times m}$ and $\mathbf{C} \in \mathbb{R}^{p\times n}$ for some $m, p \geq 1$. The transfer function, $\mathbf{G}(s)$, in (3.12) becomes matrix valued. A reduced order system analogous to (1.2) is sought with the same number of inputs m and outputs p, but with lower state space dimension $r \ll n$. If $\mathbf{V}_r \in \mathbb{R}^{n\times r}$ and $\mathbf{W}_r \in \mathbb{R}^{n\times r}$ such that $\mathbf{W}_r^T\mathbf{V}_r$ is nonsingular, we can define a (matrix-valued) reduced order transfer function $\mathbf{G}_r(s) = \mathbf{C}_r(s\mathbf{I} - \mathbf{A}_r)^{-1}\mathbf{B}_r$ with

$$\mathbf{A}_r = (\mathbf{W}_r^T\mathbf{V}_r)^{-1}\mathbf{W}_r^T\mathbf{A}\mathbf{V}_r, \quad \mathbf{B}_r = (\mathbf{W}_r^T\mathbf{V}_r)^{-1}\mathbf{W}_r^T\mathbf{B}, \quad \text{and} \quad \mathbf{C}_r = \mathbf{C}\mathbf{V}_r.$$

In order to assess "closeness" of MIMO systems, there is a natural extension of the Hilbert space, $\mathcal{H}_2$, to $p \times m$ matrix-valued functions. In particular, if $\mathbf{G}(s)$ and $\mathbf{H}(s)$ are $p \times m$ matrix-valued transfer functions associated with real stable MIMO dynamical systems, then the associated $\mathcal{H}_2$ inner product is

$$\langle \mathbf{G},\ \mathbf{H}\rangle_{\mathcal{H}_2} \stackrel{\text{def}}{=} \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathsf{tr}\left(\overline{\mathbf{G}(\imath\omega)}\,\mathbf{H}^T(\imath\omega)\right) d\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathsf{tr}\left(\mathbf{G}(-\imath\omega)\mathbf{H}^T(\imath\omega)\right) d\omega, \tag{3.13}$$

where "$\mathsf{tr}(\mathbf{M})$" denotes the trace of the matrix $\mathbf{M}$. The $\mathcal{H}_2$ norm is then

$$\|\mathbf{G}\|_{\mathcal{H}_2} \stackrel{\text{def}}{=} \left(\frac{1}{2\pi} \int_{-\infty}^{+\infty} \|\mathbf{G}(\imath\omega)\|_F^2 \, d\omega \right)^{1/2}, \tag{3.14}$$

where $\|\mathbf{F}\|_F \stackrel{\text{def}}{=} \left(\sum_{ij} |F_{ij}|^2 \right)^{1/2}$ denotes the usual Frobenius matrix norm. As before, if $\mathbf{G}(s)$ and $\mathbf{H}(s)$ represent real dynamical systems, then $\langle \mathbf{G},\ \mathbf{H} \rangle_{\mathcal{H}_2} = \langle \mathbf{H},\ \mathbf{G} \rangle_{\mathcal{H}_2}$ and $\langle \mathbf{G},\ \mathbf{H} \rangle_{\mathcal{H}_2}$ is real.

Necessary conditions for $\mathcal{H}_2$ optimality built on structured orthogonality paralleling the results of section 3.1 can be derived in this setting as well. In particular, the residue form for the inner product is a straightforward analogue of Lemma 2.4 and leads naturally to interpolation conditions. If $\mathbf{F}(s)$ is a matrix-valued meromorphic function with a pole at λ, then $\mathbf{F}(s)$ has a Laurent expansion (with matrix coefficients), and its residue, $\mathsf{res}[\mathbf{F}(s), \lambda]$, will be the coefficient matrix associated with the expansion term $(s-\lambda)^{-1}$. For example, suppose that $\mathbf{F}(s)$ has the realization $\mathbf{F}(s) = \widetilde{\mathbf{C}}\big(s\mathbf{I} - \widetilde{\mathbf{A}}\big)^{-1}\widetilde{\mathbf{B}}$. If λ is a simple pole of $\mathbf{F}(s)$, then we can assume that λ is a simple eigenvalue of $\widetilde{\mathbf{A}}$ associated with a rank-1 spectral projector $\mathbf{E}_\lambda$ and then $\mathbf{F}(s) = \frac{1}{s-\lambda}\mathbf{E}_\lambda + \mathbf{D}(s)$, where $\mathbf{D}(s)$ is analytic at $s = \lambda$, and $\mathsf{res}[\mathbf{F}(s), \lambda] = \lim_{s\to\lambda}(s-\lambda)\mathbf{F}(s) = \widetilde{\mathbf{C}}\mathbf{E}_\lambda\widetilde{\mathbf{B}}$. If λ is a double pole, then we can assume that λ is a double eigenvalue of $\widetilde{\mathbf{A}}$ associated with a rank-2 spectral projector $\mathbf{E}_\lambda$ and a rank-1 nilpotent matrix $\mathbf{N}_\lambda$ such that $\widetilde{\mathbf{A}}\mathbf{E}_\lambda = \lambda\mathbf{E}_\lambda + \mathbf{N}_\lambda$. Then $\mathbf{F}(s) = \frac{1}{(s-\lambda)^2}\mathbf{N}_\lambda + \frac{1}{(s-\lambda)}\mathbf{E}_\lambda + \mathbf{D}(s)$, where $\mathbf{D}(s)$ is analytic at $s = \lambda$, and so $\mathsf{res}[\mathbf{F}(s), \lambda] = \lim_{s\to\lambda} \frac{d}{ds}\left[(s-\lambda)^2\mathbf{F}(s)\right] = \widetilde{\mathbf{C}}\mathbf{E}_\lambda\widetilde{\mathbf{B}}$.

LEMMA 3.5. [illegible] $\mathbf{G}(s)$ [illegible] $\lambda_1, \lambda_2, \ldots, \lambda_n$ [illegible] $\mathbf{H}(s)$ [illegible] $\widetilde{\lambda}_1, \widetilde{\lambda}_2, \ldots, \widetilde{\lambda}_{\tilde{n}}$ [illegible]

$$\langle \mathbf{G},\ \mathbf{H} \rangle_{\mathcal{H}_2} = \sum_{k=1}^{\tilde{n}} \mathsf{tr}\left(\mathsf{res}[\mathbf{G}(-s)\mathbf{H}^T(s), \widetilde{\lambda}_k] \right). \tag{3.15}$$

[illegible] $\mathbf{H}(s)$ [illegible] $\mathbf{H}(s) = \widetilde{\mathbf{C}}(s\mathbf{I} - \widetilde{\mathbf{A}})^{-1}\widetilde{\mathbf{B}}$

- $\widetilde{\lambda}_k$ [illegible] $\mathbf{H}(s)$ [illegible] $\widetilde{\lambda}_k$ [illegible] $\widetilde{\mathbf{A}}$ $\widetilde{\mathbf{y}}_k$ [illegible] $\widetilde{\mathbf{x}}_k$ [illegible]

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{x}}_k = \widetilde{\lambda}_k\,\widetilde{\mathbf{x}}_k, \qquad \widetilde{\mathbf{y}}_k^*\widetilde{\mathbf{A}} = \widetilde{\lambda}_k\,\widetilde{\mathbf{y}}_k^*, \qquad \text{[illegible]} \qquad \widetilde{\mathbf{y}}_k^*\widetilde{\mathbf{x}}_k = 1,$$

[illegible]

$$\mathsf{tr}\left(\mathsf{res}[\mathbf{G}(-s)\mathbf{H}^T(s), \widetilde{\lambda}_k] \right) = \widetilde{\mathbf{c}}_k^T\mathbf{G}(-\widetilde{\lambda}_k)\widetilde{\mathbf{b}}_k,$$

[illegible] $\widetilde{\mathbf{b}}_k^T = \widetilde{\mathbf{y}}_k^*\widetilde{\mathbf{B}}$ [illegible] $\widetilde{\mathbf{c}}_k = \widetilde{\mathbf{C}}\widetilde{\mathbf{x}}_k$

- $\widetilde{\lambda}_k$ [illegible] $\mathbf{H}(s)$ [illegible] $\widetilde{\lambda}_k$ [illegible] $\widetilde{\mathbf{y}}_k$ [illegible] $\widetilde{\mathbf{x}}_k$, $\widetilde{\mathbf{A}}$ [illegible] $\widetilde{\mathbf{z}}_k$ [illegible] $\widetilde{\mathbf{w}}_k$ [illegible]

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{x}}_k = \widetilde{\lambda}_k\,\widetilde{\mathbf{x}}_k, \quad \widetilde{\mathbf{A}}\widetilde{\mathbf{w}}_k = \widetilde{\lambda}_k\,\widetilde{\mathbf{w}}_k + \widetilde{\mathbf{x}}_k, \quad \widetilde{\mathbf{y}}_k^*\widetilde{\mathbf{A}} = \widetilde{\lambda}_k\,\widetilde{\mathbf{y}}_k^*, \quad \widetilde{\mathbf{z}}_k^*\widetilde{\mathbf{A}} = \widetilde{\lambda}_k\,\widetilde{\mathbf{z}}_k^* + \widetilde{\mathbf{y}}_k^*,$$
$$\text{[illegible]} \quad \widetilde{\mathbf{y}}_k^*\widetilde{\mathbf{x}}_k = 0, \quad \widetilde{\mathbf{z}}_k^*\widetilde{\mathbf{w}}_k = 0, \quad \text{[illegible]} \quad \widetilde{\mathbf{z}}_k^*\widetilde{\mathbf{x}}_k = \widetilde{\mathbf{y}}_k^*\widetilde{\mathbf{w}}_k = 1,$$

[illegible]

$$\mathsf{tr}\left(\mathsf{res}[\mathbf{G}(-s)\mathbf{H}^T(s), \widetilde{\lambda}_k] \right) = \widetilde{\mathbf{d}}_k^T\mathbf{G}(-\widetilde{\lambda}_k)\widetilde{\mathbf{b}}_k + \widetilde{\mathbf{c}}_k^T\mathbf{G}(-\widetilde{\lambda}_k)\widetilde{\mathbf{e}}_k - \widetilde{\mathbf{c}}_k^T\mathbf{G}'(-\widetilde{\lambda}_k)\widetilde{\mathbf{b}}_k,$$

[illegible] $\widetilde{\mathbf{b}}_k$ [illegible] $\widetilde{\mathbf{c}}_k$ [illegible] $\widetilde{\mathbf{e}}_k^T = \widetilde{\mathbf{z}}_k^*\widetilde{\mathbf{B}}$ [illegible] $\widetilde{\mathbf{d}}_k = \widetilde{\mathbf{C}}\widetilde{\mathbf{w}}_k$

Now assume that $\mathbf{G}_r$ is an optimal reduced order model minimizing $\|\mathbf{G}-\mathbf{G}_r\|_{\mathcal{H}_2}$ in the sense described in (3.1) and suppose further that $\mathbf{G}_r$ has simple poles $\widetilde{\lambda}_i$. Take $\mathbf{H}(s)=\mathbf{G}_r$ in (3.5) so that $\mathbf{G}_r(s)=\sum_k \frac{1}{s-\widetilde{\lambda}_k}\widetilde{\mathbf{c}}_k\widetilde{\mathbf{b}}_k^T$ and the residue of $\mathbf{G}_r(s)$ at $\widetilde{\lambda}_k$ is matrix valued and rank one: $\mathsf{res}[\mathbf{G}_r(s),\widetilde{\lambda}_k]=\widetilde{\mathbf{c}}_k\widetilde{\mathbf{b}}_k^T$. An analysis paralleling what we have carried out above yields analogous error expressions (see also [2]) and first-order necessary conditions for the MIMO optimal $\mathcal{H}_2$ reduction problem:

$$\begin{aligned}\mathbf{G}(-\widetilde{\lambda}_k)\widetilde{\mathbf{b}}_k &= \mathbf{G}_r(-\widetilde{\lambda}_k)\widetilde{\mathbf{b}}_k,\\ \widetilde{\mathbf{c}}_k^T\mathbf{G}(-\widetilde{\lambda}_k) &= \widetilde{\mathbf{c}}_k^T\mathbf{G}_r(-\widetilde{\lambda}_k), \quad \text{and}\\ \widetilde{\mathbf{c}}_k^T\mathbf{G}'(-\widetilde{\lambda}_k)\widetilde{\mathbf{b}}_k &= \widetilde{\mathbf{c}}_k^T\mathbf{G}_r'(-\widetilde{\lambda}_k)\widetilde{\mathbf{b}}_k, \quad \text{for } k=1,\ldots,r.\end{aligned} \tag{3.16}$$

The SISO ($m=p=1$) conditions are replaced in the MIMO case by left tangential, right tangential, as well as bi-tangential interpolation conditions. From the discussion of section 2.1, if $\mathrm{Ran}(\mathbf{V}_r)$ contains $(\widetilde{\lambda}_k\mathbf{I}+\mathbf{A})^{-1}\mathbf{B}\widetilde{\mathbf{b}}_k$ and $\mathrm{Ran}(\mathbf{W}_r)$ contains $(\widetilde{\lambda}_k\mathbf{I}+\mathbf{A})^{-T}\mathbf{C}^T\widetilde{\mathbf{c}}_k$ for each $k=1,2,\ldots,r$, then the $\mathcal{H}_2$ optimality conditions given above hold. First-order interpolatory MIMO conditions have been obtained recently in other independent works as well; see [24, 35].

3.2.2. The discrete time case. An nth-order SISO discrete-time dynamical system is defined by a set of difference equations

$$G:\begin{cases}\mathbf{x}(t+1)=\mathbf{A}\mathbf{x}(t)+\mathbf{b}\,u(t)\\ y(t)=\mathbf{c}^T\mathbf{x}(t)\end{cases}\qquad\text{or}\qquad G(z)=\mathbf{c}^T(z\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}, \tag{3.17}$$

where $t\in\mathbb{Z}$ and $\mathbf{A}\in\mathbb{R}^{n\times n}$, $\mathbf{b},\mathbf{c}\in\mathbb{R}^n$. $G(z)$ is the transfer function of the system, so that if $\hat{u}(z)$ and $\hat{y}(z)$ denote the z-transforms of $u(t)$ and $y(t)$, respectively, then $\hat{y}(z)=G(z)\hat{u}(z)$. In this case, stability of G means that $\mid\lambda_i(\mathbf{A})\mid<1$ for $i=1,\ldots,n$. Also, the h_2 norm is defined as $\|G\|_{h_2}^2=\frac{1}{2\pi}\int_0^{2\pi}\mid G(e^{\imath\theta})\mid^2 d\theta$. Model reduction for discrete-time systems is defined similarly. In this setting, interpolatory (necessary) conditions for h_2 optimality of the rth-order reduced model $G_r(z)=\mathbf{c}_r^T(z\mathbf{I}-\mathbf{A}_r)^{-1}\mathbf{b}_r$ become $G\big(1/\widetilde{\lambda}_i\big)=G_r\big(1/\widetilde{\lambda}_i\big)$ and $G'\big(1/\widetilde{\lambda}_i\big)=G_r'\big(1/\widetilde{\lambda}_i\big)$ for $i=1,\ldots,r$, where $\widetilde{\lambda}_i$ denotes the ith eigenvalue of $\mathbf{A}_r$. This is a special case of results for discrete-time MIMO systems formulated previously in [10].

3.3. Lyapunov-based $\mathcal{H}_2$ optimality conditions. In this section we briefly review the Lyapunov framework for the first-order $\mathcal{H}_2$ optimality conditions and present its connection to our structured orthogonality framework.

Given a stable SISO system $G(s)=\mathbf{c}^T(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}$, let $G_r(s)=\mathbf{c}_r^T(s\mathbf{I}-\mathbf{A}_r)^{-1}\mathbf{b}_r$ be a local minimizer of dimension r for the optimal $\mathcal{H}_2$ model reduction problem (3.1) and suppose that $G_r(s)$ has simple poles at $\widetilde{\lambda}_i$, $i=1,\ldots,r$.

It is convenient to define the error system

$$G_{err}(s)\stackrel{\text{def}}{=}G(s)-G_r(s)=\mathbf{c}_{err}^T\left(s\mathbf{I}-\mathbf{A}_{err}\right)^{-1}\mathbf{b}_{err} \tag{3.18}$$

$$\text{with}\quad \mathbf{A}_{err}=\begin{bmatrix}\mathbf{A}&0\\0&\mathbf{A}_r\end{bmatrix},\quad \mathbf{b}_{err}=\begin{bmatrix}\mathbf{b}\\\mathbf{b}_r\end{bmatrix},\quad \text{and } \mathbf{c}_{err}^T=[\mathbf{c}^T\ \ -\mathbf{c}_r^T]. \tag{3.19}$$

Let $\mathbf{P}_{err}$ and $\mathbf{Q}_{err}$ be the Gramians for the error system $G_{err}(s)$; i.e., $\mathbf{P}_{err}$ and $\mathbf{Q}_{err}$ solve

$$\mathbf{A}_{err}\mathbf{P}_{err}+\mathbf{P}_{err}\mathbf{A}_{err}^T+\mathbf{b}_{err}\mathbf{b}_{err}^T=0, \tag{3.20}$$

$$\mathbf{Q}_{err}\mathbf{A}_{err}+\mathbf{A}_{err}^T\mathbf{Q}_{err}+\mathbf{c}_{err}\mathbf{c}_{err}^T=0. \tag{3.21}$$

Partition $\mathbf{P}_{err}$ and $\mathbf{Q}_{err}$:

$$\mathbf{P}_{err} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{12}^T & \mathbf{P}_{22} \end{bmatrix}, \qquad \mathbf{Q}_{err} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{12}^T & \mathbf{Q}_{22} \end{bmatrix}, \tag{3.22}$$

where $\mathbf{P}_{11}, \mathbf{Q}_{11} \in \mathbb{R}^{n\times n}$ and $\mathbf{P}_{22}, \mathbf{Q}_{22} \in \mathbb{R}^{r\times r}$. Wilson [36] showed that the reduced order model $G_r(s) = \mathbf{c}_r^T(s\mathbf{I}-\mathbf{A}_r)^{-1}\mathbf{b}_r$ can be defined in terms of a Galerkin framework as well by taking

$$\mathbf{V}_r = \mathbf{P}_{12}\mathbf{P}_{22}^{-1} \quad \text{and} \quad \mathbf{W}_r = -\mathbf{Q}_{12}\mathbf{Q}_{22}^{-1}, \tag{3.23}$$

and the resulting reduced order model satisfies the first-order conditions of the optimal $\mathcal{H}_2$ problem. It was also shown in [36] that $\mathbf{W}_r^T\mathbf{V}_r = \mathbf{I}$. The next result states the Lyapunov-based Wilson conditions for $\mathcal{H}_2$ optimality and shows their equivalence to our structured orthogonality framework.

THEOREM 3.6. [illegible] $\mathcal{H}_2$ [illegible]

$$\mathbf{P}_{12}^T\mathbf{Q}_{12} + \mathbf{P}_{22}\mathbf{Q}_{22} = 0, \tag{3.24}$$

$$\mathbf{Q}_{12}^T\mathbf{b} + \mathbf{Q}_{22}\mathbf{b}_r = 0, \tag{3.25}$$

$$\mathbf{c}_r^T\mathbf{P}_{22} - \mathbf{c}^T\mathbf{P}_{12} = 0, \tag{3.26}$$

[illegible] 3.2 [illegible]. From (3.5), consider first the case $H_1 = 0$ and H_2 is an arbitrary transfer function with simple poles at $\widetilde{\lambda}_i$, $i = 1, \ldots, r$. Write $H_2(s) = \widetilde{\mathbf{c}}^T(s\mathbf{I} - \mathbf{A}_r)^{-1}\widetilde{\mathbf{b}}$, where $\widetilde{\mathbf{b}}$ and $\widetilde{\mathbf{c}}$ can vary arbitrarily. Then from (2.12), if, for any $\widetilde{\mathbf{b}} \neq 0$, $[\widetilde{\mathbf{P}}_1^T, \widetilde{\mathbf{P}}_2^T]^T$ solves

$$\begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{A}_r \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{P}}_1 \\ \widetilde{\mathbf{P}}_2 \end{bmatrix} + \begin{bmatrix} \widetilde{\mathbf{P}}_1 \\ \widetilde{\mathbf{P}}_2 \end{bmatrix} \mathbf{A}_r^T + \begin{bmatrix} \mathbf{b} \\ \mathbf{b}_r \end{bmatrix} \widetilde{\mathbf{b}}^T = 0, \tag{3.27}$$

we have for arbitrary $\widetilde{\mathbf{c}}$

$$\langle G - G_r,\ H_2\rangle_{\mathcal{H}_2} = [\mathbf{c}^T \ \ -\mathbf{c}_r^T] \begin{bmatrix} \widetilde{\mathbf{P}}_1 \\ \widetilde{\mathbf{P}}_2 \end{bmatrix} \widetilde{\mathbf{c}} = 0.$$

Notice that $\widetilde{\mathbf{P}}_1$ and $\widetilde{\mathbf{P}}_2$ are independent of $\widetilde{\mathbf{c}}$, so for each choice of $\widetilde{\mathbf{b}}$ we must have

$$\mathbf{c}^T\widetilde{\mathbf{P}}_1 - \mathbf{c}_r^T\widetilde{\mathbf{P}}_2 = 0.$$

For $\widetilde{\mathbf{b}} = \mathbf{b}_r$, one may check directly that $\widetilde{\mathbf{P}}_1 = \mathbf{P}_{12}$ and $\widetilde{\mathbf{P}}_2 = \mathbf{P}_{22}$ in $\mathbf{P}_{err}$ that solves (3.20) in Wilson's conditions.

Likewise, from (2.13) for each choice of $\widetilde{\mathbf{c}}$, if $[\widetilde{\mathbf{Q}}_1, \widetilde{\mathbf{Q}}_2]$ solves

$$[\widetilde{\mathbf{Q}}_1, \widetilde{\mathbf{Q}}_2] \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{A}_r \end{bmatrix} + \mathbf{A}_r^T[\widetilde{\mathbf{Q}}_1, \widetilde{\mathbf{Q}}_2] + \widetilde{\mathbf{c}}[\mathbf{c}^T,\ -\mathbf{c}_r^T] = 0, \tag{3.28}$$

then we have for every $\widetilde{\mathbf{b}}$

$$\langle G - G_r,\ H_2\rangle_{\mathcal{H}_2} = \widetilde{\mathbf{b}}^T[\widetilde{\mathbf{Q}}_1, \widetilde{\mathbf{Q}}_2] \begin{bmatrix} \mathbf{b} \\ \mathbf{b}_r \end{bmatrix} = 0.$$

Similarly to the first case, $[\widetilde{\mathbf{Q}}_1, \widetilde{\mathbf{Q}}_2]$ is independent of $\widetilde{\mathbf{b}}$, so for each choice of $\widetilde{\mathbf{c}}$ we must have

$$\widetilde{\mathbf{Q}}_1\mathbf{b} + \widetilde{\mathbf{Q}}_2\mathbf{b}_r = 0,$$

and for the particular case $\widetilde{\mathbf{c}} = -\mathbf{c}_r$, one may check directly that $\widetilde{\mathbf{Q}}_1 = \mathbf{Q}_{12}^T$ and $\widetilde{\mathbf{Q}}_2 = \mathbf{Q}_{22}$ in $\mathbf{Q}_{err}$ that solves (3.21) in Wilson's conditions. The structured orthogonality condition $\langle G - G_r,\ H\rangle_{\mathcal{H}_2} = 0$ taken over all systems $H(s)$ with the same poles as G_r leads directly to the Wilson conditions (3.25) and (3.26).

The additional orthogonality condition $\langle G - G_r,\ G_r \cdot H\rangle_{\mathcal{H}_2} = 0$ taken over all $H(s)$ with the same poles as G_r will yield the remaining Wilson condition (3.24).

Observe that

$$\begin{aligned} G_r(s)H(s) &= \mathbf{c}_r^T(s\mathbf{I} - \mathbf{A}_r)^{-1}\mathbf{b}_r\,\widetilde{\mathbf{c}}^T(s\mathbf{I} - \mathbf{A}_r)^{-1}\widetilde{\mathbf{b}} \\ &= [\mathbf{c}_r^T, 0]\left(s\mathbf{I}_{2r} - \begin{bmatrix} \mathbf{A}_r & \mathbf{b}_r\widetilde{\mathbf{c}}^T \\ 0 & \mathbf{A}_r \end{bmatrix}\right)^{-1}\begin{bmatrix} 0 \\ \widetilde{\mathbf{b}} \end{bmatrix}. \end{aligned}$$

Referring to (2.12), the condition $\langle G - G_r,\ G_r \cdot H\rangle_{\mathcal{H}_2} = 0$ leads to a Sylvester equation,

$$\begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{A}_r \end{bmatrix}\begin{bmatrix} \widetilde{\mathbf{W}}_1 & \widetilde{\mathbf{P}}_1 \\ \widetilde{\mathbf{W}}_2 & \widetilde{\mathbf{P}}_2 \end{bmatrix} + \begin{bmatrix} \widetilde{\mathbf{W}}_1 & \widetilde{\mathbf{P}}_1 \\ \widetilde{\mathbf{W}}_2 & \widetilde{\mathbf{P}}_2 \end{bmatrix}\begin{bmatrix} \mathbf{A}_r^T & 0 \\ \widetilde{\mathbf{c}}\mathbf{b}_r^T & \mathbf{A}_r^T \end{bmatrix} + \begin{bmatrix} \mathbf{b} \\ \mathbf{b}_r \end{bmatrix}[0,\ \widetilde{\mathbf{b}}^T] = 0,$$

where the use of $\widetilde{\mathbf{P}}_1$ and $\widetilde{\mathbf{P}}_2$ is intended to indicate that they solve (3.27) as well. Then

$$\langle G - G_r,\ G_r \cdot H_2\rangle_{\mathcal{H}_2} = [\mathbf{c}^T,\ -\mathbf{c}_r^T]\begin{bmatrix} \widetilde{\mathbf{W}}_1 & \widetilde{\mathbf{P}}_1 \\ \widetilde{\mathbf{W}}_2 & \widetilde{\mathbf{P}}_2 \end{bmatrix}\begin{bmatrix} \mathbf{c}_r \\ 0 \end{bmatrix} = 0.$$

Alternatively, from (2.13),

(3.29)
$$\begin{bmatrix} \widetilde{\mathbf{Q}}_1 & \widetilde{\mathbf{Q}}_2 \\ \widetilde{\mathbf{Y}}_1 & \widetilde{\mathbf{Y}}_2 \end{bmatrix}\begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{A}_r \end{bmatrix} + \begin{bmatrix} \mathbf{A}_r^T & 0 \\ \widetilde{\mathbf{c}}\mathbf{b}_r^T & \mathbf{A}_r^T \end{bmatrix}\begin{bmatrix} \widetilde{\mathbf{Q}}_1 & \widetilde{\mathbf{Q}}_2 \\ \widetilde{\mathbf{Y}}_1 & \widetilde{\mathbf{Y}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{c}_r \\ 0 \end{bmatrix}[\mathbf{c}^T,\ -\mathbf{c}_r^T] = 0$$

($\widetilde{\mathbf{Q}}_1$ and $\widetilde{\mathbf{Q}}_2$ here also solve (3.28)) and

$$\langle G - G_r,\ G_r \cdot H_2\rangle_{\mathcal{H}_2} = [0,\ \widetilde{\mathbf{b}}^T]\begin{bmatrix} \widetilde{\mathbf{Q}}_1 & \widetilde{\mathbf{Q}}_2 \\ \widetilde{\mathbf{Y}}_1 & \widetilde{\mathbf{Y}}_2 \end{bmatrix}\begin{bmatrix} \mathbf{b} \\ \mathbf{b}_r \end{bmatrix} = 0.$$

Since this last equality is true for all $\widetilde{\mathbf{b}}$, and since $\widetilde{\mathbf{Y}}_1$ and $\widetilde{\mathbf{Y}}_2$ are independent of $\widetilde{\mathbf{b}}$, we see that $\widetilde{\mathbf{Y}}_1\mathbf{b} + \widetilde{\mathbf{Y}}_2\mathbf{b}_r = 0$. We know already that $\widetilde{\mathbf{Q}}_1\mathbf{b} + \widetilde{\mathbf{Q}}_2\mathbf{b}_r = 0$, so

$$\begin{bmatrix} \widetilde{\mathbf{Q}}_1 & \widetilde{\mathbf{Q}}_2 \\ \widetilde{\mathbf{Y}}_1 & \widetilde{\mathbf{Y}}_2 \end{bmatrix}\begin{bmatrix} \mathbf{b} \\ \mathbf{b}_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Define $\left[\begin{smallmatrix} \tilde{\mathbf{Q}}_1 & \tilde{\mathbf{Q}}_2 \\ \tilde{\mathbf{Y}}_1 & \tilde{\mathbf{Y}}_2 \end{smallmatrix}\right]\left[\begin{smallmatrix} \tilde{\mathbf{P}}_1 \\ \tilde{\mathbf{P}}_2 \end{smallmatrix}\right] = \left[\begin{smallmatrix} \tilde{\mathbf{R}}_1 \\ \tilde{\mathbf{R}}_2 \end{smallmatrix}\right]$. We will show that $\widetilde{\mathbf{R}}_1 = 0$. Premultiply (3.27) by $\left[\begin{smallmatrix} \tilde{\mathbf{Q}}_1 & \tilde{\mathbf{Q}}_2 \\ \tilde{\mathbf{Y}}_1 & \tilde{\mathbf{Y}}_2 \end{smallmatrix}\right]$, postmultiply (3.29) by $\left[\begin{smallmatrix} \tilde{\mathbf{P}}_1 \\ \tilde{\mathbf{P}}_2 \end{smallmatrix}\right]$, and subtract the resulting equations to get

$$\widetilde{\mathbf{R}}_1\mathbf{A}_r^T - \mathbf{A}_r^T\widetilde{\mathbf{R}}_1 = 0 \qquad \text{and} \qquad \widetilde{\mathbf{R}}_2\mathbf{A}_r^T - \mathbf{A}_r^T\widetilde{\mathbf{R}}_2 = \widetilde{\mathbf{c}}\mathbf{b}_r^T\widetilde{\mathbf{R}}_1.$$

The first equation asserts that $\widetilde{\mathbf{R}}_1$ commutes with $\mathbf{A}_r^T$, and since $\mathbf{A}_r^T$ has distinct eigenvalues, $\widetilde{\mathbf{R}}_1$ must have the same eigenvectors as $\mathbf{A}_r^T$. Let $\widetilde{\mathbf{y}}_i$, $\widetilde{\mathbf{x}}_i$ be left and right eigenvectors of $\mathbf{A}_r$ associated with $\widetilde{\lambda}_i$ (respectively, right and left eigenvectors of $\mathbf{A}_r^T$): $\mathbf{A}_r\widetilde{\mathbf{x}}_i = \widetilde{\lambda}_i\widetilde{\mathbf{x}}_i$ and $\widetilde{\mathbf{y}}_i^T\mathbf{A}_r = \widetilde{\lambda}_i\widetilde{\mathbf{y}}_i^T$. Then $\widetilde{\mathbf{R}}_1\widetilde{\mathbf{y}}_i = d_i\widetilde{\mathbf{y}}_i$. Now premultiply the second equation by $\widetilde{\mathbf{x}}_i^T$ and postmultiply by $\widetilde{\mathbf{y}}_i$ to find

$$\begin{aligned}\widetilde{\mathbf{x}}_i^T\left(\widetilde{\mathbf{R}}_2\mathbf{A}_r^T - \mathbf{A}_r^T\widetilde{\mathbf{R}}_2\right)\widetilde{\mathbf{y}}_i &= \widetilde{\mathbf{x}}_i^T\widetilde{\mathbf{c}}\mathbf{b}_r^T\widetilde{\mathbf{R}}_1\widetilde{\mathbf{y}}_i,\\ \widetilde{\mathbf{x}}_i^T\widetilde{\mathbf{R}}_2\widetilde{\mathbf{y}}_i\widetilde{\lambda}_i - \widetilde{\lambda}_i\widetilde{\mathbf{x}}_i^T\widetilde{\mathbf{R}}_2\widetilde{\mathbf{y}}_i &= \widetilde{\mathbf{x}}_i^T\widetilde{\mathbf{c}}\mathbf{b}_r^T\widetilde{\mathbf{R}}_1\widetilde{\mathbf{y}}_i,\\ 0 &= \left(\widetilde{\mathbf{x}}_i^T\widetilde{\mathbf{c}}\right)\left(\mathbf{b}_r^T\widetilde{\mathbf{y}}_i\right)\ d_i.\end{aligned}$$

Either $d_i = 0$ or one of $\widetilde{\mathbf{x}}_i^T\widetilde{\mathbf{c}}$ and $\mathbf{b}_r^T\widetilde{\mathbf{y}}_i$ must vanish, which would then imply that either $\dim H < r$ or $\dim G_r < r$. Thus $d_i = 0$ for all $i = 1, \ldots, r$ and $\widetilde{\mathbf{R}}_1 = 0$, which proves the final Wilson condition (3.24).

The converse is omitted here since it follows in a straightforward way by reversing the preceding arguments. □

Hyland and Bernstein [22] offered conditions that are equivalent to the Wilson conditions. Suppose $G_r(s)$ defined by $\mathbf{A}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r^T$ solves the optimal $\mathcal{H}_2$ problem. Then there exist positive nonnegative matrices $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{n\times n}$ and two $n \times r$ matrices $\mathbf{F}_r$ and $\mathbf{Y}_r$ such that

$$\mathbf{PQ} = \mathbf{F}_r\mathbf{M}\mathbf{Y}_r^T, \quad \mathbf{Y}_r^T\mathbf{F}_r = \mathbf{I}_r, \tag{3.30}$$

where $\mathbf{M}$ is similar to a positive definite matrix. Then $G_r(s)$ is given by $\mathbf{A}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r^T$ with $\mathbf{A}_r = \mathbf{Y}_r^T\mathbf{A}\mathbf{F}_r$, $\mathbf{b}_r = \mathbf{Y}_r^T\mathbf{b}$, and $\mathbf{c}_r^T = \mathbf{c}^T\mathbf{Y}_r$ such that, with the skew projection $\mathbf{\Pi} = \mathbf{F}_r\mathbf{Y}_r^T$, the following conditions are satisfied:

$$\operatorname{rank}(\mathbf{P}) = \operatorname{rank}(\mathbf{Q}) = \operatorname{rank}(\mathbf{PQ}), \tag{3.31}$$

$$\mathbf{\Pi}\left[\mathbf{AP} + \mathbf{PA}^T + \mathbf{bb}^T\right] = 0, \tag{3.32}$$

$$\left[\mathbf{A}^T\mathbf{Q} + \mathbf{QA} + \mathbf{cc}^T\right]\mathbf{\Pi} = 0. \tag{3.33}$$

Note that in both [36] and [22], the first-order necessary conditions are given in terms of (coupled) Lyapunov equations. Both [36] and [22] proposed iterative algorithms to obtain a reduced order model satisfying these Lyapunov-based first-order conditions. However, the main drawback in each case is that both approaches require solving two large-scale Lyapunov equations at each step of the algorithm. [40] discusses computational issues related to solving associated linearized problems within each step.

Theorems 3.4 and 3.6 show the equivalence between the structured orthogonality conditions and Lyapunov- and interpolation-based conditions for $\mathcal{H}_2$ optimality, respectively. To complete the discussion, we formally state the equivalence between the Lyapunov and interpolation frameworks.

LEMMA 3.7 (equivalence of Lyapunov and interpolation frameworks). [illegible] [22] [illegible] (3.31)–(3.33) [illegible] [36] [illegible] (3.23) [illegible] [26] [illegible] (3.11) [illegible] [36, 22] [illegible] $\mathcal{H}_2$ [illegible]

We note that the connection between the Lyapunov and interpolation frameworks has not been observed in the literature before. This result shows that solving the optimal $\mathcal{H}_2$ problem in the Krylov framework is equivalent to solving it in the Lyapunov framework. This leads to the Krylov-based method proposed in the next section.

4. Iterated interpolation. We propose an effective numerical algorithm that produces a reduced order model $G_r(s)$ satisfying the interpolation-based first-order necessary conditions (3.11). Effectiveness of the proposed algorithm results from the fact that we use rational Krylov steps to construct a $G_r(s)$ that meets the first-order conditions (3.11). No Lyapunov solvers or dense matrix decompositions are needed. Therefore, the method is suited for large-scale systems where $n \gg 1000$.

Several approaches have been proposed in the literature to compute reduced order models that satisfy [illegible] of first-order necessary conditions; see [37, 34, 9, 21, 26, 22, 36, 25]. However, these approaches do not seem to be suitable for large-scale problems. The ones based on Lyapunov-based conditions, e.g., [36, 22, 34, 37], require solving a couple of Lyapunov equations at each step of the iteration. To our knowledge, the only methods that depend on interpolation-based necessary conditions have been proposed in [25] and [26]. The authors work directly with the transfer functions of $G(s)$ and $G_r(s)$; make an iteration on the denominator [25] or poles and residues [26] of $G_r(s)$; and explicitly compute $G(s)$, $G_r(s)$, and their derivatives at certain points in the complex plane. However, working with the transfer function, its values, and its derivative values explicitly is not desirable in large-scale settings. Indeed, one will most likely be given a state space representation of $G(s)$ rather than the transfer function. And trying to compute the coefficients of the transfer function can be highly ill-conditioned. These approaches are similar to [30, 31], where interpolation is done by explicit usage of transfer functions. On the other hand, our approach, which is detailed below, is based on the connection between interpolation and effective rational Krylov iteration, and is therefore numerically effective and stable.

Let $\boldsymbol{\sigma}$ denote the set of interpolation points $\{\sigma_1, \ldots, \sigma_r\}$; use these interpolation points to construct a reduced order model, $G_r(s)$, that interpolates both $G(s)$ and $G'(s)$ at $\{\sigma_1, \ldots, \sigma_r\}$; let $\boldsymbol{\lambda}(\boldsymbol{\sigma}) = \{\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_r\}$ denote the resulting reduced order poles of $G_r(s)$; hence $\boldsymbol{\lambda}(\boldsymbol{\sigma})$ is a function from $\mathbb{C}^r \mapsto \mathbb{C}^r$. Define the function $\mathbf{g}(\boldsymbol{\sigma}) = \boldsymbol{\lambda}(\boldsymbol{\sigma}) + \boldsymbol{\sigma}$. Note that $\mathbf{g}(\boldsymbol{\sigma}) : \mathbb{C}^r \mapsto \mathbb{C}^r$. Aside from issues related to the ordering of the reduced order poles, $\mathbf{g}(\boldsymbol{\sigma}) = \mathbf{0}$ yields $\boldsymbol{\lambda}(\boldsymbol{\sigma}) = -\boldsymbol{\sigma}$; i.e., the reduced order poles $\boldsymbol{\lambda}(\boldsymbol{\sigma})$ are mirror images of the interpolation points $\boldsymbol{\sigma}$. Hence, $\mathbf{g}(\boldsymbol{\sigma}) = \mathbf{0}$ is equivalent to (3.11) and is a necessary condition for $\mathcal{H}_2$ optimality of the reduced order model, $G_r(s)$. Thus one can formulate a search for optimal $\mathcal{H}_2$ reduced order systems by considering the root-finding problem $\mathbf{g}(\boldsymbol{\sigma}) = \mathbf{0}$. Many plausible approaches to this problem originate with Newton's method, which appears as

$$\boldsymbol{\sigma}^{(k+1)} = \boldsymbol{\sigma}^{(k)} - (\mathbf{I} + \mathbf{J})^{-1} \left(\boldsymbol{\sigma}^{(\mathbf{k})} + \boldsymbol{\lambda} \left(\boldsymbol{\sigma}^{(\mathbf{k})} \right) \right). \tag{4.1}$$

In (4.1), $\mathbf{J}$ is the usual $r \times r$ Jacobian of $\boldsymbol{\lambda}(\boldsymbol{\sigma})$ with respect to $\boldsymbol{\sigma}$: for $\mathbf{J} = [J_{i,j}]$, $J_{i,j} = \frac{\partial \widetilde{\lambda}_i}{\partial \sigma_j}$ for $i, j = 1, \ldots, r$. How to compute $\mathbf{J}$ will be clarified in section 4.3.

4.1. Proposed algorithm. We seek a reduced order transfer function $G_r(s)$ that interpolates $G(s)$ at the mirror images of the poles of $G_r(s)$ by solving the equivalent root-finding problem, say by a variant of (4.1). It is often the case that in the neighborhood of an $\mathcal{H}_2$ optimal shift set, the entries of the Jacobian matrix become small and simply setting $\mathbf{J} = 0$ might serve as a relaxed iteration strategy. This leads to a successive substitution framework: $\sigma_i \leftarrow -\lambda_i(\mathbf{A}_r)$; successive interpolation steps using a rational Krylov method are used so that at the $(i+1)$st step interpolation points are chosen as the mirror images of the Ritz values from the ith step. Despite its simplicity, this appears to be a very effective strategy in many circumstances.

Here is a sketch of the proposed algorithm.

ALGORITHM 4.1. An iterative rational Krylov algorithm (IRKA).

1. [illegible] σ_i, $i = 1, \ldots, r$ [illegible] tol
2. [illegible] $\mathbf{V}_r$ [illegible] $\mathbf{W}_r$ [illegible] $\mathrm{Ran}(\mathbf{V}_r) = \mathrm{span}\left\{(\sigma_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}, \ldots, (\sigma_r\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}\right\}$, $\mathrm{Ran}(\mathbf{W}_r) = \mathrm{span}\left\{(\sigma_1\mathbf{I} - \mathbf{A}^T)^{-1}\mathbf{c}, \ldots, (\sigma_r\mathbf{I} - \mathbf{A}^T)^{-1}\mathbf{c}\right\}$ [illegible] $\mathbf{W}_r^T\mathbf{V}_r = \mathbf{I}$
3. [illegible] ($\{\sigma_i\} >$ tol)
   (a) $\mathbf{A_r} = \mathbf{W_r^T A V_r}$
   (b) [illegible] $\sigma_i \longleftarrow -\lambda_i(\mathbf{A_r})$, $i = 1, \ldots, r$
   (c) [illegible] $\mathbf{V}_r$ [illegible] $\mathbf{W}_r$ [illegible] $\mathrm{Ran}(\mathbf{V}_r) = \mathrm{span}\left\{(\sigma_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}, \ldots, (\sigma_r\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}\right\}$, $\mathrm{Ran}(\mathbf{W}_r) = \mathrm{span}\left\{(\sigma_1\mathbf{I} - \mathbf{A}^T)^{-1}\mathbf{c}, \ldots, (\sigma_r\mathbf{I} - \mathbf{A}^T)^{-1}\mathbf{c}\right\}$, [illegible] $\mathbf{W}_r^T\mathbf{V}_r = \mathbf{I}$
4. $\mathbf{A_r} = \mathbf{W_r^T A V_r}$, $\mathbf{b}_r = \mathbf{W}_r^T\mathbf{b}$, $\mathbf{c}_r^T = \mathbf{c}^T\mathbf{V}_r$

Upon convergence, the first-order necessary conditions (3.11) for $\mathcal{H}_2$ optimality will be satisfied. Notice that step 3(b) could be replaced with some variant of a Newton step (4.1).

We have implemented the above algorithm and applied it to many different large-scale systems. In each of our numerical examples, the algorithm worked very effectively: It has always converged after a small number of steps and resulted in stable reduced systems. For those standard test problems we tried where a global optimum is known, Algorithm 4.1 converged to this global optimum.

It should be noted that the solution is obtained via Krylov projection methods only and its computation is suitable for large-scale systems. To our knowledge, this is the first numerically effective approach for the optimal $\mathcal{H}_2$ reduction problem.

We know that the reduced model $G_r(s)$ resulting from the above algorithm will satisfy the first-order optimality conditions. Moreover, from Theorem 3.1 this reduced order model is globally optimal in the following sense.

COROLLARY 4.1. [illegible] $G_r(s)$ [illegible] 4.1 [illegible] $G_r(s)$ [illegible] $G(s)$ [illegible] $\mathcal{H}_2$ [illegible] $G_r(s)$

Therefore Algorithm 4.1 generates a reduced model, $G_r(s)$, which is the optimal solution for a restricted $\mathcal{H}_2$ problem.

4.2. Initial shift selection. For the proposed algorithm, the final reduced model can depend on the initial shift selection. Nonetheless for most of the cases, a random initial shift selection resulted in satisfactory reduced models. For small-order benchmark examples taken from [22, 25, 37, 34], the algorithm converged to the global minimizer. For larger problems, the results were as good as those obtained by balanced truncation. Therefore, while staying within a numerically effective Krylov projection framework, we have been able to produce results close to or better than those obtained by balanced truncation (which requires the solution of two large-scale Lyapunov equations).

We outline some initialization strategies that can be expected to improve the results. Recall that at convergence, interpolation points are mirror images of the eigenvalues of $\mathbf{A}_r$. The eigenvalues of $\mathbf{A}_r$ might be expected to approximate the eigenvalues of $\mathbf{A}$. Hence, at convergence, interpolation points will lie in the mirror spectrum of $\mathbf{A}$. Therefore, one could choose initial shifts randomly distributed within a region containing the mirror image of the numerical range of $\mathbf{A}$. The boundary of the numerical range can be estimated by computing the eigenvalues of $\mathbf{A}$ with the smallest and largest real and imaginary parts using numerically effective tools such as the implicitly restarted Arnoldi (IRA) algorithm.

The starting point for another initialization strategy is the $\mathcal{H}_2$ expression presented in Proposition 3.3. Based on this expression, it is appropriate to initiate the proposed algorithm with $\sigma_i = -\lambda_i(\mathbf{A})$, where $\lambda_i(\mathbf{A})$ are the poles with big residues, ϕ_i for $i = 1, \ldots, r$. The main disadvantage of this approach is that it requires a modal state space decomposition for $G(s)$, which will be numerically expensive for large-scale problems. However, there might be some applications where the original state space representation is in the modal form and ϕ_i might be directly read from the entries of the matrices $\mathbf{b}$ and $\mathbf{c}^T$.

Unstable reduced order models are not acceptable candidates for optimal $\mathcal{H}_2$ reduction. Nonetheless stability of a reduced model is not guaranteed a priori and might depend on the initial shift selection. We have observed that if one avoids making extremely unrealistic initial shift selections, stability will be preserved. In our simulations we have never generated an unstable system when the initial shift selection was not drastically different from the mirror spectrum of $\mathbf{A}$, but otherwise random. We were able to produce an unstable reduced order system; however, this occurred for a case where the real parts of the eigenvalues of $\mathbf{A}$ were between -1.5668×10^{-1} and -2.0621×10^{-3}, yet we chose initial shifts bigger than 50. We believe that with a good starting point, stability will not be an issue. These considerations are illustrated for many numerical examples in section 5.

. . . 4.1. Based on the first-order conditions (3.16) discussed in section 3.2.1 for MIMO systems $\mathbf{G}(s) = \mathbf{C}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}$, one can extend IRKA to the MIMO case by replacing $(\sigma_i\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}$ with $(\sigma_i\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}\widetilde{\mathbf{b}}_i$ and $(\sigma_i\mathbf{I}-\mathbf{A}^T)^{-1}\mathbf{c}$ with $(\sigma_i\mathbf{I}-\mathbf{A})^{-1}\mathbf{C}^T\widetilde{\mathbf{c}}_i$ in Algorithm 4.1, where $\widetilde{\mathbf{b}}_i$ and $\widetilde{\mathbf{c}}_i$ are as defined in section 3.2.1.

. . . 4.2. In the discrete-time case described in (3.17) above, the root-finding problem becomes $\mathbf{g}(\boldsymbol{\sigma}) = \boldsymbol{\Sigma}\boldsymbol{\lambda}(\boldsymbol{\sigma}) - \mathbf{e}$, where $\mathbf{e}^T = [1, 1, \ldots, 1]$ and $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\sigma})$. Therefore, for discrete-time systems, step 3(b) of Algorithm 4.1 becomes $\sigma_i \leftarrow 1/\lambda_i(\mathbf{A_r})$ for $i = 1, \ldots, r$. Moreover, the associated Newton step is

$$\boldsymbol{\sigma}^{(k+1)} = \boldsymbol{\sigma}^{(k)} - (\mathbf{I} + \boldsymbol{\Lambda}^{-1}\boldsymbol{\Sigma}\mathbf{J})^{-1}\left(\boldsymbol{\sigma}^{(k)} - \boldsymbol{\Lambda}^{-1}\mathbf{e}\right),$$

where $\Lambda = \mathrm{diag}(\boldsymbol{\lambda})$.

4.3. A Newton framework for IRKA. As discussed above, Algorithm 4.1 uses the successive substitution framework by simply setting $\mathbf{J} = \mathbf{0}$ in the Newton step (4.1). The Newton framework for IRKA can be easily obtained by replacing step 3(b) of Algorithm 4.1 with the Newton step (4.1). The only point to clarify for the Newton framework is the computation of the Jacobian, which measures the sensitivity of the reduced system poles with respect to shifts.

Given $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$, suppose that σ_i, $i = 1, \ldots, r$, are r distinct points in $\mathbb{C}$, none of which are eigenvalues of $\mathbf{A}$, and define the complex r-tuple $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_r]^T \in \mathbb{C}^r$ together with related matrices:

$$\mathbf{V}_r(\boldsymbol{\sigma}) = \begin{bmatrix} (\sigma_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} & (\sigma_2\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} & \ldots & (\sigma_r\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} \end{bmatrix} \in \mathbb{C}^{n\times r} \tag{4.2}$$

and

$$\mathbf{W}_r^T(\boldsymbol{\sigma}) = \begin{bmatrix} \mathbf{c}^T(\sigma_1\mathbf{I} - \mathbf{A})^{-1} \\ \mathbf{c}^T(\sigma_2\mathbf{I} - \mathbf{A})^{-1} \\ \vdots \\ \mathbf{c}^T(\sigma_r\mathbf{I} - \mathbf{A})^{-1} \end{bmatrix} \in \mathbb{C}^{r\times n}. \tag{4.3}$$

We normally suppress the dependence on $\boldsymbol{\sigma}$ and write $\mathbf{V}_r(\boldsymbol{\sigma}) = \mathbf{V}_r$ and $\mathbf{W}_r(\boldsymbol{\sigma}) = \mathbf{W}_r$. Hence, the reduced order system matrix $\mathbf{A}_r$ is given by $\mathbf{A}_r = (\mathbf{W}_r^T\mathbf{V}_r)^{-1}\mathbf{W}_r^T\mathbf{A}\mathbf{V}_r$, where $(\mathbf{W}_r^T\mathbf{V}_r)^{-1}\mathbf{W}_r$ plays the role of $\mathbf{W}_r$ in Algorithm 4.1. Let $\widetilde{\lambda}_i$, for $i = 1, \ldots, r$, denote the eigenvalues of $\mathbf{A}_r$. Hence, the Jacobian computation amounts to computing $\mathbf{J}(i,j) = \frac{\partial\widetilde{\lambda}_i}{\partial\sigma_j}$. The following result shows how to compute the Jacobian for the Newton formulation of the IRKA method proposed here.

LEMMA 4.2. [illegible] $\mathbf{A}_r = (\mathbf{W}_r^T\mathbf{V}_r)^{-1}\mathbf{W}_r^T\mathbf{A}\mathbf{V}_r$ [illegible] $\widetilde{\lambda}_i$ [illegible] $|\widetilde{\mathbf{x}}_i^T\mathbf{W}_r^T\mathbf{V}_r\widetilde{\mathbf{x}}_i| = 1$ [illegible] $\mathbf{W}_r^T\mathbf{A}\mathbf{V}_r\widetilde{\mathbf{x}}_i = \widetilde{\lambda}_i\mathbf{W}_r^T\mathbf{V}_r\widetilde{\mathbf{x}}_i$ [illegible]

$$\frac{\partial\widetilde{\lambda}_i}{\partial\sigma_j} = \widetilde{\mathbf{x}}_i^T\partial_j\mathbf{W}_r^T\left(\mathbf{A}\mathbf{V}_r\widetilde{\mathbf{x}}_i - \widetilde{\lambda}_i\mathbf{V}_r\widetilde{\mathbf{x}}_i\right) + \left(\widetilde{\mathbf{x}}_i^T\mathbf{W}_r^T\mathbf{A} - \widetilde{\lambda}_i\widetilde{\mathbf{x}}_i^T\mathbf{W}_r^T\right)\partial_j\mathbf{V}_r\widetilde{\mathbf{x}}_i, \tag{4.4}$$

[illegible] $\partial_j\mathbf{W}_r^T = \frac{\partial}{\partial\sigma_j}\mathbf{W}_r^T = -\mathbf{e}_j\,\mathbf{c}(\sigma_j\mathbf{I}-\mathbf{A})^{-2}$ and $\partial_j\mathbf{V}_r = \frac{\partial}{\partial\sigma_j}\mathbf{V}_r = -(\sigma_j\mathbf{I}-\mathbf{A})^{-2}\mathbf{b}\,\mathbf{e}_j^T$. [illegible]. With $\mathbf{V}_r(\boldsymbol{\sigma}) = \mathbf{V}_r$ and $\mathbf{W}_r(\boldsymbol{\sigma}) = \mathbf{W}_r$ defined as in (4.2) and (4.3), both $\mathbf{W}_r^T\mathbf{A}\mathbf{V}_r$ and $\mathbf{W}_r^T\mathbf{V}_r$ are complex symmetric matrices. Write $\widetilde{\lambda}$ for $\widetilde{\lambda}_i$ and $\widetilde{\mathbf{x}}$ for $\widetilde{\mathbf{x}}_i$, so

$$\text{(a) } \mathbf{W}_r^T\mathbf{A}\mathbf{V}_r\widetilde{\mathbf{x}} = \widetilde{\lambda}\mathbf{W}_r^T\mathbf{V}_r\widetilde{\mathbf{x}} \quad\text{and}\quad \text{(b) } \widetilde{\mathbf{x}}^T\mathbf{W}_r^T\mathbf{A}\mathbf{V}_r = \widetilde{\lambda}\,\widetilde{\mathbf{x}}^T\mathbf{W}_r^T\mathbf{V}_r. \tag{4.5}$$

Equation (4.5b) is obtained by transposition of (4.5a). $\widetilde{\mathbf{x}}^T\mathbf{W}_r^T\mathbf{V}_r$ is a left eigenvector for $\mathbf{A}_r$ associated with $\widetilde{\lambda}_i$. Differentiate (4.5a) with respect to σ_j, premultiply with $\widetilde{\mathbf{x}}^T$, and simplify using (4.5b):

$$\widetilde{\mathbf{x}}^T\partial_j\mathbf{W}_r^T\left(\mathbf{A}\mathbf{V}_r\widetilde{\mathbf{x}} - \widetilde{\lambda}\mathbf{V}_r\widetilde{\mathbf{x}}\right) + \left(\widetilde{\mathbf{x}}^T\mathbf{W}_r^T\mathbf{A} - \widetilde{\lambda}\widetilde{\mathbf{x}}^T\mathbf{W}_r^T\right)\partial_j\mathbf{V}_r\widetilde{\mathbf{x}} = \left(\frac{\partial\widetilde{\lambda}}{\partial\sigma_j}\right)\widetilde{\mathbf{x}}^T\mathbf{W}_r^T\mathbf{V}_r\widetilde{\mathbf{x}},$$

where $\partial_j\mathbf{W}_r^T = \frac{\partial}{\partial\sigma_j}\mathbf{W}_r^T = \mathbf{e}_j\,\mathbf{c}^T(\sigma_j\mathbf{I}-\mathbf{A})^{-2}$ and $\partial_j\mathbf{V}_r = \frac{\partial}{\partial\sigma_j}\mathbf{V} = (\sigma_j\mathbf{I}-\mathbf{A})^{-2}\mathbf{b}\,\mathbf{e}_j^T$. This completes the proof. ∎

5. Numerical examples. We first compare our approach with the earlier approaches [22, 25, 37] on [illegible] benchmark examples presented in those papers. We show that in each case we attain the minimum, the main difference being that we achieve this minimum in a numerically efficient way. For each low-order model, comparisons are made using data taken from the original sources [22, 25, 37]. We then test our method in large-scale settings.

5.1. Low-order models and comparisons. Consider the following 4 models:

- FOM-1: Example 6.1 in [22]. State space representation of FOM-1 is given by

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & -150 \\ 1 & 0 & 0 & -245 \\ 0 & 1 & 0 & -113 \\ 0 & 0 & 1 & -19 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 4 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

  We reduce the order to $r = 3, 2, 1$ using the proposed successive rational Krylov algorithm, denoted by IRKA, and compare our results with the gradient flow method of [37], denoted by GFM; the orthogonal projection method of [22], denoted by OPM; and the balanced truncation method, denoted by BTM.

- FOM-2: Example in [25]. Transfer function of FOM-2 is given by

$$G(s) = \frac{2s^6 + 11.5s^5 + 57.75s^4 + 178.625s^3 + 345.5s^2 + 323.625s + 94.5}{s^7 + 10s^6 + 46s^5 + 130s^4 + 239s^3 + 280s^2 + 194s + 60}.$$

We reduce the order to $r = 6, 5, 4, 3$ using IRKA and compare our results with GFM, OPM, BTM, and the method proposed in [25], denoted by LMPV.

- FOM-3: Example 1 in [34]. Transfer function of FOM-3 is given by

$$G(s) = \frac{s^2 + 15s + 50}{s^4 + 5s^3 + 33s^2 + 79s + 50}.$$

We reduce the order to $r = 3, 2, 1$ using IRKA and compare our results with GFM, OPM, BTM, and the method proposed in [34], denoted by SMM.

- FOM-4: Example 2 in [34]. Transfer function of FOM-4 is given by

$$G(s) = \frac{10000s + 5000}{s^2 + 5000s + 25}.$$

We reduce the order to $r = 1$ IRKA and compare our results with GFM, OPM, BTM, and SMM.

For all these cases, the resulting relative $\mathcal{H}_2$ errors $\frac{\|G(s)-G_r(s)\|_{\mathcal{H}_2}}{\|G(s)\|_{\mathcal{H}_2}}$ are tabulated in Table 5.1 below, which clearly illustrates that the proposed method is the only one that attains the minimum in each case. More importantly, the proposed method achieves this value in a numerically efficient way staying in the Krylov projection framework. No Lyapunov solvers or dense matrix decompositions are needed. The

TABLE 5.1
Comparison.

Model	r	IRKA	GFM	OPM
FOM-1	1	4.2683×10^{-1}	4.2709×10^{-1}	4.2683×10^{-1}
FOM-1	2	3.9290×10^{-2}	3.9299×19^{-2}	3.9290×10^{-2}
FOM-1	3	1.3047×10^{-3}	1.3107×19^{-3}	1.3047×10^{-3}
FOM-2	3	1.171×10^{-1}	1.171×10^{-1}	Divergent
FOM-2	4	8.199×10^{-3}	8.199×10^{-3}	8.199×10^{-3}
FOM-2	5	2.132×10^{-3}	2.132×10^{-3}	Divergent
FOM-2	6	5.817×10^{-5}	5.817×10^{-5}	5.817×10^{-5}
FOM-3	1	4.818×10^{-1}	4.818×10^{-1}	4.818×10^{-1}
FOM-3	2	2.443×10^{-1}	2.443×10^{-1}	Divergent
FOM-3	3	5.74×10^{-2}	5.98×10^{-2}	5.74×10^{-2}
FOM-4	1	9.85×10^{-2}	9.85×10^{-2}	9.85×10^{-2}

Model	r	BTM	LMPV	SMM
FOM-1	1	4.3212×10^{-1}		
FOM-1	2	3.9378×10^{-2}		
FOM-1	3	1.3107×10^{-3}		
FOM-2	3	2.384×10^{-1}	1.171×10^{-1}	
FOM-2	4	8.226×10^{-3}	8.199×10^{-3}	
FOM-2	5	2.452×10^{-3}	2.132×10^{-3}	
FOM-2	6	5.822×10^{-5}	2.864×10^{-4}	
FOM-3	1	4.848×10^{-1}		4.818×10^{-1}
FOM-3	2	3.332×10^{-1}		2.443×10^{-1}
FOM-3	3	5.99×10^{-2}		5.74×10^{-2}
FOM-4	1	9.949×10^{-1}	9.985×10^{-2}	

only arithmetic operations involved are LU decompositions and some linear solvers. Moreover, our method does not require starting from an initial balanced realization, as suggested in [37] and [22]. In all these simulations, we have chosen a random initial shift selection, and the algorithm converged in a small number of steps.

To illustrate the evolution of the $\mathcal{H}_2$ error throughout the iteration, consider the model FOM-2 with $r = 3$. The proposed method yields the following third-order optimal reduced model:

$$G_3(s) = \frac{2.155s^2 + 3.343s + 33.8}{s^3 + 7.457s^2 + 10.51s + 17.57}.$$

Poles of $G_3(s)$ are $\widetilde{\lambda}_1 = -6.2217$ and $\widetilde{\lambda}_{2,3} = -6.1774 \times 10^{-1} \pm \imath 1.5628$, and it can be shown that $G_3(s)$ interpolates the first two moments of $G(s)$ at $-\widetilde{\lambda}_i$ for $i = 1, 2, 3$. Hence, the first-order interpolation conditions are satisfied. This also means that if we start Algorithm 4.1 with the mirror images of these Ritz values, the algorithm converges at the first step. However, we will try four random, but , , initial selections. In other words, we start away from the optimal solution. We test the following four selections: $\mathcal{S}_1 = \{-1.01,\ -2.01,\ -30000\}$, $\mathcal{S}_2 = \{0,\ 10,\ 3\}$, $\mathcal{S}_3 = \{1,\ 10,\ 3\}$, and $\mathcal{S}_4 = \{0.01,\ 20,\ 10000\}$. With selection $\mathcal{S}_1$, we have initiated the algorithm with some negative shifts close to system poles, and consequently with a relative $\mathcal{H}_2$ error bigger than 1. However, in all four cases including $\mathcal{S}_1$, the algorithm converged in 5 steps to the same reduced model. The results are depicted in Figure 5.1.

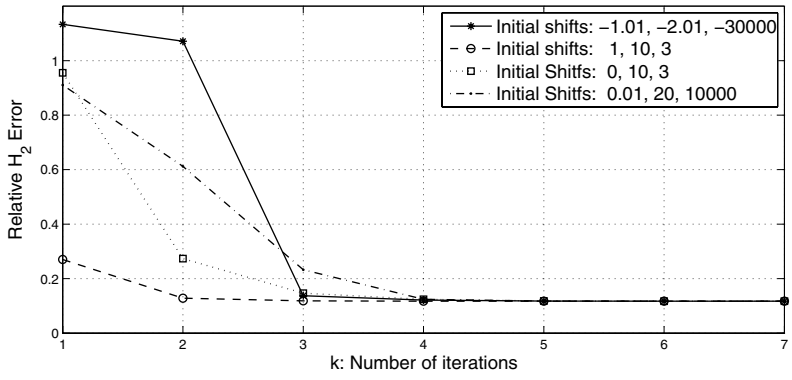


FIG. 5.1. *$\mathcal{H}_2$ norm of the error system vs. the number of iterations.*

Before testing the proposed method in large-scale settings, we investigate FOM-4 further. As pointed out in [34], since $r = 1$, the optimal $\mathcal{H}_2$ problem can be formulated as only a function of the reduced system pole. It was shown in [34] that there are two local minima: (i) one corresponding to a reduced pole at -0.0052 and consequently a reduced order model $G_1^l(s) = \frac{1.0313}{s+0.0052}$ and a relative error of 0.9949, and (ii) one to a reduced pole at -4998 and consequently a reduced model $G_1^g = \frac{9999}{s+4998}$ with a relative error of 0.0985. It follows that the latter, i.e., $G_1^g(s)$, is the global minimum. The first-order balanced truncation for FOM-4 can be easily computed as $G_1^b(s) = \frac{1.0308}{s+0.0052}$. Therefore, it is highly likely that if one starts from a balanced realization, the algorithm would converge to the local minimum $G_1^l(s)$. This was indeed the case as reported in [34]. SMM converged to the local minimum for all starting poles bigger than -0.47. On the other hand, SMM converged to the

global minimum when it was started with an initial pole smaller than -0.47. We have observed exactly the same situation in our simulations. When we start from an initial shift selection smaller than 0.48, IRKA converged to the local minimum. However, when we start with any initial shift bigger than 0.48, the algorithm converged to the global minimum in at most 3 steps. Therefore, for this example we were not able the avoid the local minimum if we started from a [illegible] shift. These observations perfectly agree with the discussion of section 4.2. Note that the transfer function of FOM-4 can be written as

$$G(s) = \frac{10000s + 5000}{s^2 + 5000s + 25} = \frac{0.99}{s + 0.0050} + \frac{9999}{s + 5000}.$$

The pole at -5000 is the one corresponding to the large residue of 9999. Therefore, a good initial shift is 5000. And if we start the proposed algorithm with an initial shift at 5000, or close, the algorithm converges to the global minimum.

5.2. CD player example. The original model describes the dynamics between a lens actuator and the radial arm position in a portable CD player. The model has 120 states, i.e., n = 120, with a single input and a single output. As illustrated in [4], the Hankel singular values of this model do not decay rapidly and hence the model is relatively hard to reduce. Moreover, even though the Krylov-based methods resulted in good local behavior, they are observed to yield large $\mathcal{H}_\infty$ and $\mathcal{H}_2$ error compared to balanced truncation.

We compare the performance of the proposed method, Algorithm 4.1, with that of balanced truncation. Balanced truncation is well known to lead to small $\mathcal{H}_\infty$ and $\mathcal{H}_2$ error norms; see [4, 19]. This is due mainly to global information available through the two system Gramians, the reachability and observability Gramians, which are each solutions of a different Lyapunov equation. We reduce the order to r, with r varying from 2 to 40; and for each r value, we compare the $\mathcal{H}_2$ error norms due to balanced truncation and due to Algorithm 4.1. For the proposed algorithm, two different selections have been tried for the initial shifts. (1) Mirror images of the eigenvalues corresponding to large residuals, and (2) a random selection with real parts in the interval $[10^{-1}, 10^3]$ and the imaginary parts in the interval $[1, 10^5]$. To make this selection, we looked at the poles of $G(s)$ having the maximum/minimum real and imaginary parts. The results showing the relative $\mathcal{H}_2$ error for each r are depicted in Figure 5.2. The figure reveals that both selection strategies work quite well. Indeed, the random initial selection behaves better than the residual-based selection and outperforms balanced truncation for almost all the r values except $r = 2, 24, 36$. However, even for these r values, the resulting $\mathcal{H}_2$ error is not far away from the one due to balanced truncation. For the range $r = [12, 22]$, the random selection clearly outperforms the balanced truncation. We would like to emphasize that these results were obtained by a [illegible] shift selection and staying in the numerically effective Krylov projection framework [illegible] requiring any solutions to large-scale Lyapunov equations. This is the main difference our proposed algorithm has with existing methods and what makes it numerically effective in large-scale settings.

To examine convergence behavior, we reduce the order to $r = 8$ and $r = 10$ using Algorithm 4.1. At each step of the iteration, we compute the $\mathcal{H}_2$ error due to the current estimate and plot this error versus the iteration index. The results are shown in Figure 5.3. The figure illustrates two important properties for both cases $r = 8$ and $r = 10$: (1) At each step of the iteration, the $\mathcal{H}_2$ norm of the error is reduced. (2) The algorithm converges after 3 steps. The resulting reduced models are stable for both cases.

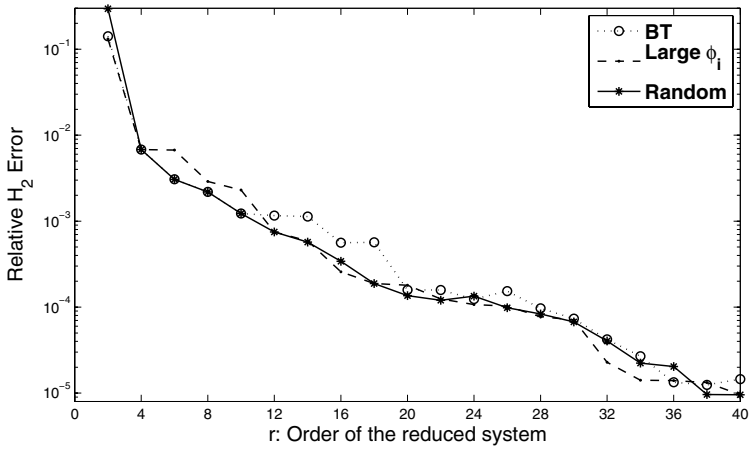


FIG. 5.2. *Relative $\mathcal{H}_2$ norm of the error system vs. r.*

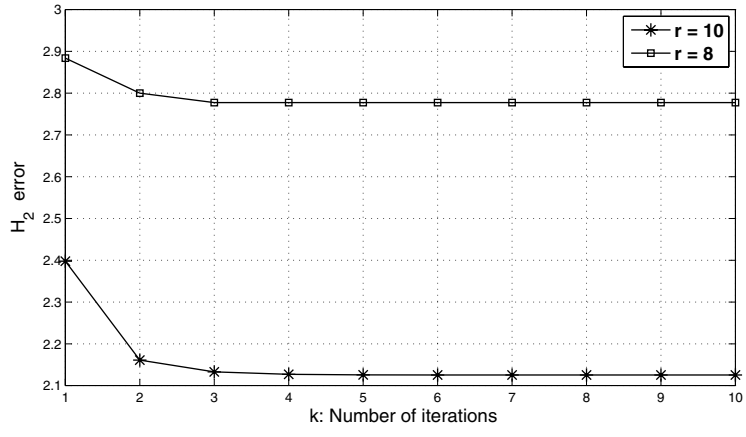


FIG. 5.3. *$\mathcal{H}_2$ norm of the error system vs. the number of iterations.*

5.3. A semidiscretized heat transfer problem for optimal cooling of steel profiles. This problem arises during a cooling process in a rolling mill when different steps in the production process require different temperatures of the raw material. To achieve high throughput, one seeks to reduce the temperature as fast as possible to the required level before entering the next production phase. This is realized by spraying cooling fluids on the surface and must be controlled so that material properties, such as durability or porosity, stay within given quality standards. The problem is modeled as boundary control of a two-dimensional heat equation. A finite element discretization using two steps of mesh refinement with maximum mesh width of 1.382×10^{-2} results in a system of the form

$$\mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t), \qquad y(t) = \mathbf{c}^T\mathbf{x}(t),$$

with state dimension $n = 20209$, i.e., $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{20209\times 20209}$, $\mathbf{b} \in \mathbb{R}^{20209\times 7}$, $\mathbf{c}^T \in \mathbb{R}^{6\times 20209}$. Note that in this case $\mathbf{E} \neq \mathbf{I}$, but the algorithm works with the obvious

modifications. For details regarding the modeling, discretization, optimal control design, and model reduction for this example, see [29, 7, 8]. We consider the full-order SISO system that associates the sixth input of this system with the second output. We apply our algorithm and reduce the order to $r = 6$. Amplitude Bode plots of $G(s)$ and $G_r(s)$ are shown in Figure 5.4. The output response of $G_r(s)$ is virtually indistinguishable from $G(s)$ in the frequency range considered. IRKA converged in 7 iteration steps in this case, although some interpolation points converged in the first 2–3 steps. The relative $\mathcal{H}_\infty$ error obtained with our sixth order system was 7.85×10^{-3}. Note that in order to apply balanced truncation in this example, one would need to solve [illegible] Lyapunov equations (since $\mathbf{E} \neq \mathbf{I}$) of order 20209. This presents a severe computational challenge, though there have been interesting approaches to addressing it (e.g., [5]).

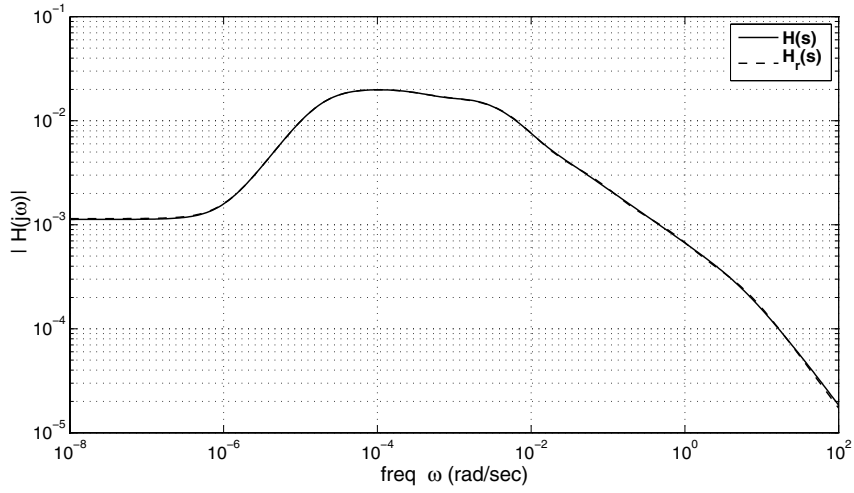


FIG. 5.4. *Amplitude Bode plots of $H(s)$ and $H_r(s)$.*

5.4. Successive substitution vs. Newton framework. In this section, we present two examples to show the effect of the Newton formulation for IRKA on two low-order examples.

The first example is FOM-1 from section 5.1. For this example, for reduction to $r = 1$, the optimal shift is $\sigma = 0.4952$. We initiate both iterations, successive substitution and Newton frameworks, away from this optimal value with an initial selection $\sigma_0 = 10^4$. Figure 5.5 illustrates how each process converges. As the figure shows, even though it takes almost 15 iterations with oscillations for the successive substitution framework to converge, the Newton formulation reaches the optimal shift in 4 steps.

The second example in this section is a third-order model with a transfer function

$$G = \frac{-s^2 + (7/4)s + 5/4}{s^3 + 2s^2 + (17/16)s + 15/32}.$$

One can exactly compute the optimal $\mathcal{H}_2$ reduced model for $r = 1$ as

$$G_r(s) = \frac{0.97197}{s + 0.2727272}$$

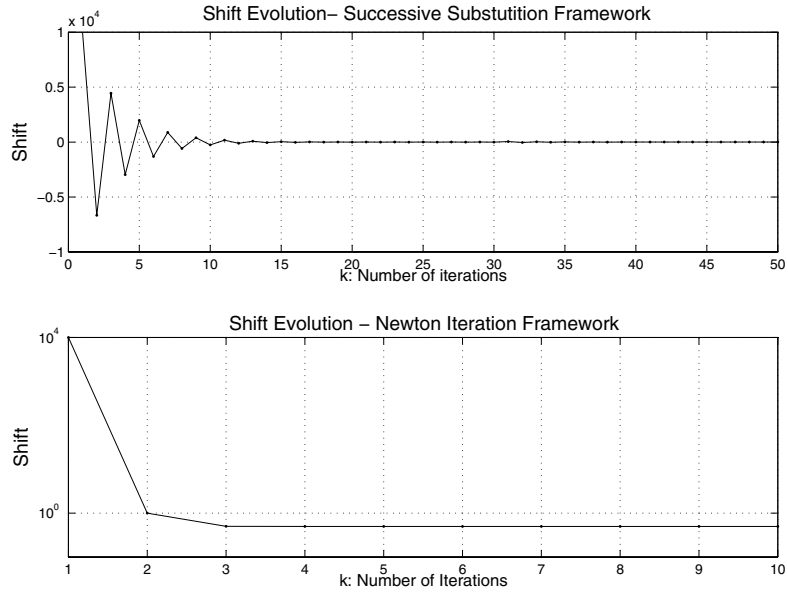


FIG. 5.5. *Comparison for* FOM-1.

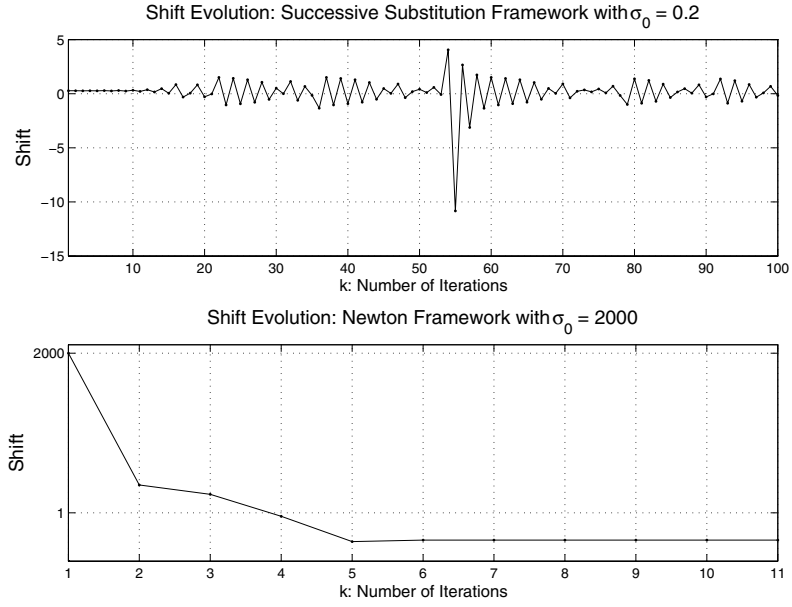


FIG. 5.6. *Comparison for the random third-order model.*

and easily show that this reduced model interpolates $G(s)$ and its derivative at $\sigma = 0.2727272$. We initiate Algorithm 4.1 with $\sigma_0 = 0.27$, very close to the optimal shift. We initiate the Newton framework at $\sigma_0 = 2000$, far away from the optimal solution. Convergence behavior of both models is depicted in Figure 5.6. The figure shows that for this example, the successive substitution framework is divergent and indeed

$\frac{\partial\tilde{\lambda}}{\partial\sigma} \approx 1.3728$. On the other hand, the Newton framework is able to converge to the optimal solution in a small number of steps.

6. Conclusions. We have developed an interpolation-based rational Krylov algorithm that iteratively corrects interpolation locations until first-order $\mathcal{H}_2$ optimality conditions are satisfied. The resulting method proves numerically effective and well suited for large-scale problems. A new derivation of the interpolation-based necessary conditions is presented and shown to be equivalent to two other common frameworks for $\mathcal{H}_2$ optimality.

Appendix.

LEMMA A.1. [illegible] $\mathbf{M}$ [illegible]

$$\int_{-\infty}^{+\infty} (\imath\omega - \mathbf{M})^{-1}\, d\omega \stackrel{def}{=} \lim_{L\to\infty} \int_{-L}^{L} (\imath\omega - \mathbf{M})^{-1}\, d\omega = \pi\mathbf{I}.$$

[illegible] Observe that for any $L > 0$,

$$\int_{-L}^{L} (\imath\omega - \mathbf{M})^{-1}\, d\omega = \int_{-L}^{L} (-\imath\omega - \mathbf{M})(\omega^2 + \mathbf{M}^2)^{-1}\, d\omega = \int_{-L}^{L} (-\mathbf{M})(\omega^2\mathbf{I} + \mathbf{M}^2)^{-1}\, d\omega.$$

Fix a contour Γ contained in the open left half plane so that the interior of Γ contains all eigenvalues of $\mathbf{M}$. Then

$$-\mathbf{M}(\omega^2\mathbf{I} + \mathbf{M}^2)^{-1} = \frac{1}{2\pi\imath} \int_{\Gamma} \frac{-z}{\omega^2 + z^2} (z\mathbf{I} - \mathbf{M})^{-1}\, dz.$$

For any fixed value z in the left half plane,

$$\text{P.V.} \int_{-\infty}^{+\infty} \frac{d\omega}{\imath\omega - z} = \lim_{L\to\infty} \int_{-L}^{L} \frac{-z}{\omega^2 + z^2}\, d\omega = \pi.$$

Thus,

$$\begin{aligned}\lim_{L\to\infty} \int_{-L}^{L} (-\mathbf{M})(\omega^2\mathbf{I} + \mathbf{M}^2)^{-1}\, d\omega &= \frac{1}{2\pi\imath} \int_{\Gamma} \lim_{L\to\infty} \left(\int_{-L}^{L} \frac{-z}{\omega^2 + z^2}\, d\omega \right) (z\mathbf{I} - \mathbf{M})^{-1}\, dz \\ &= \frac{1}{2\pi\imath} \int_{\Gamma} \pi\, (z\mathbf{I} - \mathbf{M})^{-1}\, dz = \pi\mathbf{I}. \qquad \square\end{aligned}$$

REFERENCES

[1] A.C. ANTOULAS, *Recursive modeling of discrete-time time series*, in Linear Algebra for Control Theory, P. Van Dooren and B. W. Wyman, eds., IMA Vol. Math. Appl. 62, Springer-Verlag, New York, 1993, pp. 1–20.

[2] A.C. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, Adv. Des. Control 6, SIAM, Philadelphia, 2005.

[3] A.C. ANTOULAS AND J.C. WILLEMS, *A behavioral approach to linear exact modeling*, IEEE Trans. Automat. Control, 38 (1993), pp. 1776–1802.

[4] A.C. ANTOULAS, D.C. SORENSEN, AND S. GUGERCIN, *A survey of model reduction methods for large scale systems*, in Structured Matrices in Mathematics, Computer Science, and Engineering, I (Boulder, CO, 1999), Contemp. Math. 280, AMS, Providence, RI, 2001, pp. 193–219.

[5] J. BADIA, P. BENNER, R. MAYO, E.S. QUINTANA-ORTÍ, G. QUINTANA-ORTÍ, AND J. SAAK, *Parallel order reduction via balanced truncation for optimal cooling of steel profiles*, in Proceedings of the 11th International European Conference on Parallel Processing, EuroPar 2005, Lisbon, J. C. Cunha and P. D. Medeiros, eds., Lecture Notes in Comput. Sci. 3648, Springer-Verlag, Berlin, 2005, pp. 857–866.

[6] L. Baratchart, M. Cardelli, and M. Olivi, *Identification and rational ℓ_2 approximation: A gradient algorithm*, Automat., 27 (1991), pp. 413–418.
[7] P. Benner, *Solving large-scale control problems*, IEEE Control Systems Mag., 24 (2004), pp. 44–59.
[8] P. Benner and J. Saak, *Efficient numerical solution of the LQR-problem for the heat equation*, Proc. Appl. Math. Mech., 4 (2004), pp. 648–649.
[9] A.E. Bryson and A. Carrier, *Second-order algorithm for optimal model order reduction*, J. Guidance Control Dynam., 13 (1990), pp. 887–892.
[10] A. Bunse-Gerstner, D. Kubalinska, G. Vossen, and D. Wilczek, *H_2-optimal Model Reduction for Large Scale Discrete Dynamical MIMO Systems*, ZeTeM Technical Report 07-04, University of Bremen, 2007; available online from http://www.math.uni-bremen.de/zetem/reports/reports-liste.html#reports2007.
[11] C. De Villemagne and R. Skelton, *Model reduction using a projection formulation*, Internat. J. Control, 40 (1987), pp. 2141–2169.
[12] P. Feldman and R.W. Freund, *Efficient linear circuit analysis by Padé approximation via a Lanczos method*, IEEE Trans. Computer-Aided Design, 14 (1995), pp. 639–649.
[13] P. Fulcheri and M. Olivi, *Matrix rational H_2 approximation: A gradient algorithm based on Schur analysis*, SIAM J. Control Optim., 36 (1998), pp. 2103–2127.
[14] D. Gaier, *Lectures on Complex Approximation*, Birkhäuser, Boston, 1987.
[15] K. Gallivan, E. Grimme, and P. Van Dooren, *A rational Lanczos algorithm for model reduction*, Numer. Algorithms, 2 (1996), pp. 33–63.
[16] K. Gallivan, P. Van Dooren, and E. Grimme, *On some recent developments in projection-based model reduction*, in ENUMATH 97 (Heidelberg), World Scientific, River Edge, NJ, 1998, pp. 98–113.
[17] E.J. Grimme, *Krylov Projection Methods for Model Reduction*, Ph.D. thesis, University of Illinois, Urbana-Champaign, Urbana, IL, 1997.
[18] S. Gugercin, *Projection Methods for Model Reduction of Large-Scale Dynamical Systems*, Ph.D. thesis, Rice University, Houston, TX, 2002.
[19] S. Gugercin and A.C. Antoulas, *A comparative study of 7 model reduction algorithms*, in Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, 2000, pp. 2367–2372.
[20] S. Gugercin and A.C. Antoulas, *An $\mathcal{H}_2$ error expression for the Lanczos procedure*, in Proceedings of the 42nd IEEE Conference on Decision and Control, 2003, pp. 1869–1872.
[21] Y. Halevi, *Frequency weighted model reduction via optimal projection*, in Proceedings of the 29th IEEE Conference on Decision and Control, 1990, pp. 2906–2911.
[22] D.C. Hyland and D.S. Bernstein, *The optimal projection equations for model reduction and the relationships among the methods of Wilson, Skelton, and Moore*, IEEE Trans. Automat. Control, 30 (1985), pp. 1201–1211.
[23] J.G. Korvink and E.B. Rudyni, *Oberwolfach benchmark collection*, in Dimension Reduction of Large-Scale Systems, P. Benner, G. Golub, V. Mehrmann, and D. Sorensen, eds., Lecture Notes in Comput. Sci. Engrg. 45, Springer-Verlag, New York, 2005.
[24] D. Kubalinska, A. Bunse-Gerstner, G. Vossen, and D. Wilczek, *H_2-optimal interpolation based model reduction for large-scale systems*, in Proceedings of the 16th International Conference on System Science, Wroclaw, Poland, 2007.
[25] A. Lepschy, G.A. Mian, G. Pinato, and U. Viaro, *Rational L_2 approximation: A non-gradient algorithm*, in Proceedings of the 30th IEEE Conference on Decision and Control, 1991, pp. 2321–2323.
[26] L. Meier and D.G. Luenberger, *Approximation of linear constant systems*, IEEE Trans. Automat. Control, 12 (1967), pp. 585–588.
[27] B.C. Moore, *Principal component analysis in linear system: Controllability, observability and model reduction*, IEEE Trans. Automat. Control, 26 (1981), pp. 17–32.
[28] C.T. Mullis and R.A. Roberts, *Synthesis of minimum roundoff noise fixed point digital filters*, IEEE Trans. Circuits Systems, CAS-23 (1976), pp. 551–562.
[29] T. Penzl, *Algorithms for model reduction of large dynamical systems*, Linear Algebra Appl., 415 (2006), pp. 322–343.
[30] L.T. Pillage and R.A. Rohrer, *Asymptotic waveform evaluation for timing analysis*, IEEE Trans. Computer-Aided Design, 9 (1990), pp. 352–366.
[31] V. Raghavan, R.A. Rohrer, L.T. Pillage, J.Y. Lee, J.E. Bracken, and M.M. Alaybeyi, *AWE inspired*, in Proceedings of the IEEE Custom Integrated Circuits Conference, 1993, pp. 18.1.1–18.1.8.
[32] F. Riesz and B. Sz.-Nagy, *Functional Analysis*, Ungar, New York, 1955.
[33] A. Ruhe, *Rational Krylov algorithms for nonsymmetric eigenvalue problems* II: *Matrix pairs*, Linear Algebra Appl., 197 (1994), pp. 283–295.

[34] J.T. Spanos, M.H. Milman, and D.L. Mingori, *A new algorithm for $\mathcal{L}_2$ optimal model reduction*, Automat., 28 (1992), pp. 897–909.

[35] P. Van Dooren, K.A. Gallivan, and P.-A. Absil, *H_2 optimal model reduction of MIMO systems*, Appl. Math. Lett., to appear; doi:10.1016/j.aml.2007.09.015.

[36] D.A. Wilson, *Optimum solution of model reduction problem*, Proc. IEE-D, 117 (1970), pp. 1161–1165.

[37] W.-Y. Yan and J. Lam, *An approximate approach to $\mathcal{H}_2$ optimal model reduction*, IEEE Trans. Automat. Control, 44 (1999), pp. 1341–1358.

[38] A. Yousouff and R.E. Skelton, *Covariance equivalent realizations with applications to model reduction of large-scale systems*, in Control and Dynamic Systems, Vol. 22, C. T. Leondes, ed., Academic Press, New York, 1985, pp. 273–348.

[39] A. Yousouff, D.A. Wagie, and R.E. Skelton, *Linear system approximation via covariance equivalent realizations*, J. Math. Anal. Appl., 196 (1985), pp. 91–115.

[40] D. Zigic, L. Watson, and C. Beattie, *Contragredient transformations applied to the optimal projection equations*, Linear Algebra Appl., 188/189 (1993), pp. 665–676.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 639–656

ANALYTIC RESULTS FOR THE EIGENVALUES OF CERTAIN TRIDIAGONAL MATRICES*

ALLAN R. WILLMS†

Abstract. The eigenvalue problem for a certain tridiagonal matrix with complex coefficients is considered. The eigenvalues and eigenvectors are shown to be expressible in terms of solutions of a certain scalar trigonometric equation. Explicit solutions of this equation are obtained for several special cases, and further analysis of this equation in several other cases provides information about the distribution of eigenvalues.

Key words. eigenvalues, eigenvectors, tridiagonal matrices, analytic solutions

AMS subject classification. 15A18

DOI. 10.1137/070695411

1. Introduction.

Recently Yueh [9], Kouachi [6], and da Fonseca [4] have studied the eigenvalues of certain tridiagonal matrices, developing a trigonometric equation whose solution yields the eigenvalues. In several special cases this equation has explicit solutions, and exact expressions for the eigenvalues and eigenvectors were obtained. In this paper, we extend their work by more completely describing one further special case where an explicit solution is possible and by analyzing this equation in a number of further cases where information about the distribution of eigenvalues can be obtained.

Consider the tridiagonal matrix

$$A = \begin{bmatrix} -\alpha + b & c_1 & & & \\ a_1 & b & c_2 & & \\ & a_2 & \ddots & \ddots & \\ & & \ddots & b & c_{n-1} \\ & & & a_{n-1} & -\beta + b \end{bmatrix} \tag{1}$$

with the restriction

$$\sqrt{a_i c_i} = d \neq 0, \quad 1 \leq i < n. \tag{2}$$

All variables are, in general, complex. Without loss of generality we may assume that d is the principal square root of $a_i c_i$ so that its argument is in the range $(-\pi/2, \pi/2]$. The matrix A has $n+3$ (complex) degrees of freedom: a_i, $1 \leq i < n$, b, d, α, and β (c_i being determined by a_i and d). However, since the eigenvalues of the matrix $M + bI$ are equal to $\lambda_i + b$, where λ_i are the eigenvalues of the matrix M, it follows that the eigenvalues of A will be of the form $\lambda = b + f(a_i, c_i, \alpha, \beta)$. It is also not difficult to see that the quantities a_i and c_i always occur as a product in the characteristic polynomial for A, so that the eigenvalues are in fact of the form $\lambda = b + f(d, \alpha, \beta)$. Thus the eigenvalues of A have four complex degrees of freedom. Since b simply appears in an

*Received by the editors June 25, 2007; accepted for publication (in revised form) by M. Chu February 5, 2008; published electronically June 13, 2008.
http://www.siam.org/journals/simax/30-2/69541.html
†Department of Mathematics and Statistics, University of Guelph, Guelph, ON N1G 2W1, Canada (AWillms@uoguelph.ca).

additive way, we will concentrate on $\mathbb{C}^3$, the space of parameters (d, α, β). We identify the complex plane with $\mathbb{R}^2$, and thus the matrix parameter space is six-dimensional. Most of the special cases yielding explicit solutions for the eigenvalues of A are two-dimensional manifolds in this space, although one case is four-dimensional. In addition we will obtain information about the distribution of eigenvalues in various other three- and four-dimensional manifolds in this space.

Yueh [9] considered matrices of the form A but with constant subdiagonal and superdiagonal entries, that is, $a_i = a$ and $c_i = c$, $1 \le i < n$. He established that the eigenvalues of the $n \times n$ matrix A are of the form $\lambda = b + 2\sqrt{ac}\cos\theta$, and the eigenvectors can also be given in terms of θ, where θ is a solution of

$$(3)\quad ac\sin\big((n+1)\theta\big) + (\alpha+\beta)\sqrt{ac}\sin(n\theta) + \alpha\beta\sin\big((n-1)\theta\big) = 0, \qquad \theta \ne k\pi,\ k \in \mathbb{Z}.$$

He solved this equation explicitly in several special cases:

1. $\alpha = \beta = 0$. In this case the matrix A is Toeplitz, and the analytic expressions for the eigenvalues and eigenvectors are well known [8, p. 514] to be

$$\lambda_m = b + 2\sqrt{ac}\cos\left(\frac{m\pi}{n+1}\right), \quad v_j^{(m)} = \left(\frac{a}{c}\right)^{j-1}\sin\left(\frac{jm\pi}{n+1}\right), \quad 1 \le m \le n,$$

2. $\alpha = 0$, $\beta = \pm\sqrt{ac}$ (or switch α and β),
3. $\alpha = \beta = \pm\sqrt{ac}$,
4. $\alpha = -\beta = \pm\sqrt{ac}$.

The resulting expressions for the eigenvalues and eigenvectors in the last three cases are similar in flavor to the first case [9]. He did not explicitly solve for the special case $\alpha = -c$, $\beta = -a$. But, by using (3), since $\sin((n+1)\theta)+\sin((n-1)\theta) = 2\sin(n\theta)\cos\theta$, for this case we get

$$\sin(n\theta) = 0, \quad \text{or} \quad 2\sqrt{ac}\cos\theta - (a+c) = 0,$$

which results in eigenvalues

$$\lambda_m = b + 2\sqrt{ac}\cos\left(\frac{m\pi}{n}\right), \quad 1 \le m < n, \quad \text{and} \quad \lambda_n = b + a + c.$$

For $b = -(a+c)$ this result was known at least as far back as 1954 [7, pp. 365–366].

Kouachi [6] used a different method to study eigenvalues and eigenvectors of the matrix A and generalized to the case where the off-diagonal entries satisfy

$$a_i c_i = \begin{cases} d_1^2 & \text{if } i \text{ is odd,} \\ d_2^2 & \text{if } i \text{ is even.} \end{cases}$$

As in Yueh's work, the eigenvalues are given in terms of $\cos\theta$ and a nonlinear equation involving θ is specified. In addition to the special cases considered by Yueh, Kouachi also found explicit expressions for the eigenvalues and eigenvectors in the case $\alpha\beta = d_2^2$, n even. This amounts to a generalization of Ledermann and Reuter's [7] result, although only for even n, and, unfortunately, there are a number of substantial typographical errors in Kouachi's paper particularly in the exposition of this case.

The eigenvalues of certain perturbed tridiagonal k-Toeplitz matrices were recently studied by da Fonseca [4]. In particular, da Fonesca gives a trigonometric relation satisfied by the eigenvalues of a 2-Toeplitz matrix with perturbed entries in the top left and bottom right corners. This result is more general than those of Yueh and

Kouachi, and those presented here, since the matrix A and those studied by Yueh and Kouachi are special types of 2-Toeplitz matrices. However, da Fonseca gives an explicit formula for the eigenvalues in just one special case of the perturbation parameters (α and β) but no others. (Incidentally, there is a typographical error in his paper for this formula also: $b_1^2 + b_2^2$ on page 65 should be $b_1 + b_2$.)

We also note another recent result by da Fonseca [5] which relates the eigenvalues of the matrix

$$C = [\mu \min\{i, j\} - \nu]_{i,j=1,\dots,n}, \qquad \mu > 0, \quad \mu \neq \nu,$$

to those of a matrix of the form A (1) with $a_i = c_i = -1$, $i = 1, \dots, n-1$, $b = 2$, $\alpha = 1 - \mu/(\mu - \nu)$, and $\beta = 1$. However, he obtains explicit results only for a couple of special values for μ and ν, which turn out to be two of the special cases that we consider here.

In this paper we provide a general expression for the eigenvalues and eigenvectors of the matrix A, which generalizes Yueh's result and is contained within Kouachi's and da Fonseca's. We enumerate ,, of the special cases where this general expression yields explicit results for the eigenvalues and eigenvectors; particularly, we describe the case $\alpha\beta = d^2$ fully, extending the work of Yueh and Kouachi. In addition, we provide analysis in several other special cases not considered by any of the above three authors where explicit solutions are not possible but information on the distribution of eigenvalues can be obtained.

2. General results. Similar to Yueh [9], we utilize the methods of symbolic calculus of semi-infinite sequences [3]. The pertinent facts are as follows. The convolution of two sequences $x = \{x_j\}_{j=0}^{\infty}$ and $y = \{y_j\}_{j=0}^{\infty}$ is the sequence $z = \{z_n\}_{n=0}^{\infty}$ whose nth component is

$$z_n = \sum_{j=0}^{n} x_j y_{n-j}.$$

Convolution is a symmetric operation $xy = yx$, distributes over addition $x(y + z) = xy + xz$, and is associative with respect to scalar multiplication $x(cy) = c(xy)$ for any scalar c. We define $I = \{1, 0, \dots\}$ and $S = \{0, 1, 0, \dots\}$. Thus

$$Ix = x \quad \text{and} \quad Sx = \{0, x_0, x_1, \dots\},$$

and, in particular,

$$S\{x_{j+1}\}_{j=0}^{\infty} = \{0, x_1, x_2, \dots\} = x - x_0 I$$

and

$$S^2\{x_{j+2}\}_{j=0}^{\infty} = \{0, 0, x_2, x_3, \dots\} = x - x_0 I - x_1 S.$$

Each sequence x with $x_0 \neq 0$ has a unique inverse y such that $xy = I$, where $y_0 = 1/x_0$ and y_n, $n > 0$, is defined recursively by $y_n = -(\sum_{j=1}^{n} x_j y_{n-j})/x_0$. In particular, the inverse of $(S - \gamma I)$, for scalar γ, is

$$(S - \gamma I)^{-1} = \left\{ \frac{-1}{\gamma^{j+1}} \right\}_{j=0}^{\infty}. \tag{4}$$

Consider now the eigenproblem for the matrix A, $Av = \lambda v$, which can be written as follows:

$$\begin{aligned}(b-\lambda)v_1 + c_1 v_2 &= \alpha v_1,\\ a_1 v_1 + (b-\lambda)v_2 + c_2 v_3 &= 0,\\ &\vdots\\ a_{n-2}v_{n-2} + (b-\lambda)v_{n-1} + c_{n-1}v_n &= 0,\\ a_{n-1}v_{n-1} + (b-\lambda)v_n &= \beta v_n.\end{aligned} \tag{5}$$

Since $d^2 = a_j c_j$, for all $j = 1, \ldots, n-1$, it follows that $d/c_j = a_j/d$. Define $q_1 = 1$ and q_j, $1 < j \le n$, by the recursion

$$q_j = \frac{d}{c_{j-1}} q_{j-1} = \frac{a_{j-1}}{d} q_{j-1}, \qquad 1 < j \le n. \tag{6}$$

Formally, the q_j are given by

$$q_j = \left(\frac{1 a_1 a_2 \cdots a_{j-1}}{1 c_1 c_2 \cdots c_{j-1}} \right)^{1/2}; \tag{7}$$

however, the specific branch of the square root function that needs to be used for each subsequent j is not always the principal branch but rather is determined by the requirement that $q_j = d q_{j-1}/c_{j-1}$. Note that, by (2), a_j and c_j, and hence q_j, are all nonzero so that the vector u given by

$$v_j = q_j u_j, \quad 1 \le j \le n, \tag{8}$$

is well defined. Substitute (8) into system (5), dividing the jth equation by q_j. By using (6) this process yields the system

$$\begin{aligned}(b-\lambda)u_1 + d u_2 &= \alpha u_1,\\ d u_1 + (b-\lambda)u_2 + d u_3 &= 0,\\ &\vdots\\ d u_{n-2} + (b-\lambda)u_{n-1} + d u_n &= 0,\\ d u_{n-1} + (b-\lambda)u_n &= \beta u_n.\end{aligned} \tag{9}$$

We now extend the vector u to a semi-infinite sequence $\{u_j\}_{j=0}^{\infty}$ and impose $u_0 = 0$ and $u_{n+1} = 0$. System (9) can then be written as

$$d\,\{u_{j+2}\}_{j=0}^{\infty} + (b-\lambda)\,\{u_{j+1}\}_{j=0}^{\infty} + d\,\{u_j\}_{j=0}^{\infty} = \{f_{j+1}\}_{j=0}^{\infty}, \tag{10}$$

where f is the sequence defined by

$$f_j = \begin{cases} \alpha u_1 & \text{if } j = 1,\\ \beta u_n & \text{if } j = n,\\ 0 & \text{otherwise.}\end{cases} \tag{11}$$

Taking the convolution of (10) with S^2 gives

$$d\,(u - u_0 I - u_1 S) + (b-\lambda) S\,(u - u_0 I) + d S^2 u = S\,(f - f_0 I),$$

and by using the facts that $u_0 = 0$ and $f_0 = 0$ we have

$$\big(dS^2 + (b-\lambda)S + dI\big)u = (f + du_1 I)\, S. \tag{12}$$

Let

$$\gamma_\pm = \frac{1}{2d}\left[-(b-\lambda) \pm \sqrt{\omega}\right], \tag{13}$$

where $\omega = (b-\lambda)^2 - 4d^2$, be the two roots of $dx^2 + (b-\lambda)x + d = 0$. Note that $\gamma_+\gamma_- = 1$, and thus we may write $\gamma_\pm = e^{\pm i\theta}$, where $\theta \in \mathbb{C}$ by taking $\theta = \arg\gamma_+ - i\ln|\gamma_+|$. Since $-(b-\lambda)/d = \gamma_+ + \gamma_- = e^{i\theta} + e^{-i\theta} = 2\cos\theta$, the eigenvalues are given by

$$\lambda = b + 2d\cos\theta. \tag{14}$$

Further, since $\cos(x+iy) = \cosh(y)\cos(a) - i\sinh(b)\sin(a)$, it follows that the eigenvalue is real either if θ is real or if $\mathrm{Re}(\theta) = k\pi$, $k \in \mathbb{Z}$. With this notation, (12) becomes

$$d(S - \gamma_+ I)(S - \gamma_- I)u = (f + du_1 I)\, S,$$

and from (4) we obtain

$$\begin{aligned}
u &= \left\{\frac{-1}{\gamma_+^{j+1}}\right\}_{j=0}^{\infty} \left\{\frac{-1}{\gamma_-^{j+1}}\right\}_{j=0}^{\infty} \left(\frac{1}{d}f + u_1 I\right) S \\
&= \left\{\sum_{j=0}^{m} \frac{1}{\gamma_+^{j+1}\gamma_-^{m-j+1}}\right\}_{m=0}^{\infty} \left(\frac{1}{d}f + u_1 I\right) S \\
&= \left\{\sum_{j=0}^{m} \gamma_-^{j}\gamma_+^{m-j}\right\}_{m=0}^{\infty} \left(\frac{1}{d}f + u_1 I\right) S \\
&= \begin{cases} \left\{\dfrac{\gamma_+^{m+1} - \gamma_-^{m+1}}{\gamma_+ - \gamma_-}\right\}_{m=0}^{\infty} \left(\dfrac{1}{d}f + u_1 I\right) S & \text{if } \gamma_+ \neq \gamma_-, \\ \left\{(m+1)\gamma_+^m\right\}_{m=0}^{\infty} \left(\dfrac{1}{d}f + u_1 I\right) S & \text{if } \gamma_+ = \gamma_-, \end{cases} \\
&= \begin{cases} \left\{\dfrac{\sin((m+1)\theta)}{\sin\theta}\right\}_{m=0}^{\infty} \left(\dfrac{1}{d}f + u_1 I\right) S & \text{if } \gamma_+ \neq \gamma_-, \\ \left\{(m+1)e^{im\theta}\right\}_{m=0}^{\infty} \left(\dfrac{1}{d}f + u_1 I\right) S & \text{if } \gamma_+ = \gamma_-. \end{cases}
\end{aligned} \tag{15}$$

If we define the function $g : \mathbb{Z} \times \mathbb{C} \to \mathbb{C}$ as

$$g(n,\theta) = \begin{cases} \dfrac{\sin(n\theta)}{\sin\theta} & \text{if } \theta \neq k\pi,\ k \in \mathbb{Z}, \\ n & \text{if } \theta = 2k\pi,\ k \in \mathbb{Z}, \\ (-1)^{n-1}n & \text{if } \theta = (2k+1)\pi,\ k \in \mathbb{Z}, \end{cases} \tag{16}$$

then g is continuous in θ, and, since $\gamma_+ = \gamma_-$ is equivalent to $\theta = k\pi$, $k \in \mathbb{Z}$, both of the cases in (15) collapse to

$$u = \left\{ g(m+1,\theta) \right\}_{m=0}^{\infty} \left(\frac{1}{d} f + u_1 I \right) S. \tag{17}$$

Computing the convolutions in this last expression and noting from (11) that the only nonzero entries of the sequence f are $f_1 = \alpha u_1$ and $f_n = \beta u_n$ yields $u_0 = 0$ and

$$u_j = u_1 \left[g(j,\theta) + \frac{\alpha}{d} g(j-1,\theta) \right] + H(j-n-1) \frac{\beta}{d} u_n g(j-n,\theta), \qquad j \geq 1, \tag{18}$$

where $H(x)$ is the unit step function: $H(x) = 1$ if $x \geq 0$, and $H(x) = 0$ if $x < 0$. By using (18) with $j = n+1$, the condition $u_{n+1} = 0$ becomes

$$\left[g(n+1,\theta) + \frac{\alpha}{d} g(n,\theta) \right] u_1 + \frac{\beta}{d} u_n = 0, \tag{19}$$

where we have used the fact that $g(1,\theta) = 1$. Finally, by using (18) at $j = n$ and noting that u_1 cannot be zero (otherwise, the vector v must be zero) we obtain the necessary and sufficient condition for $\lambda = b + 2d\cos\theta$ to be an eigenvalue of A, namely,

$$g(n+1,\theta) + \frac{\alpha+\beta}{d} g(n,\theta) + \frac{\alpha\beta}{d^2} g(n-1,\theta) = 0. \tag{20}$$

Equations (20) and (14) correspond to those derived by Yueh; however, he multiplied (20) by $d^2 \sin\theta$ to clear the denominator and dealt with the case $\theta = k\pi$ separately.

We now show that there are exactly n solutions (counting multiplicity) of (20) in the region

$$R = \{ \theta = (x+iy) \mid 0 \leq x \leq \pi,\ x, y \in \mathbb{R} \}, \tag{21}$$

where roots on the boundary of R are counted with half weight. The continuous function g is 2π-periodic in θ and is an even function of θ. Consequently once the roots of (20) in R are found, all roots of (20) can be determined. Setting $\xi = e^{i\theta}$ and multiplying (20) by the nonzero quantity $d^2\xi^{n+1}/\xi$ gives

$$\frac{d^2(\xi^{2n+2} - 1) + d(\alpha+\beta)\xi(\xi^{2n} - 1) + \alpha\beta\xi^2(\xi^{2n-2} - 1)}{\xi^2 - 1} = 0. \tag{22}$$

Clearly $\xi = \pm 1$ are roots of the numerator; hence, $\xi^2 - 1$ divides the numerator, and we are left with a $2n$th order polynomial in ξ which necessarily has $2n$ roots, some possibly repeated. Since $e^{i\theta}$ is 2π-periodic, it follows that there are $2n$ roots of (20) in the region $R \cup \hat{R}$, where $\hat{R} = \{\theta \mid -\pi < \mathrm{Re}(\theta) < 0\}$. By the even property of g, every root in $\mathrm{int}(R)$ (the interior of R) has a corresponding root in $\hat{R}$. Since R and $\hat{R}$ are disjoint, it follows that twice the number or roots in $\mathrm{int}(R)$ plus the number of roots on the boundary of R is $2n$. Further, roots on the left boundary of R of the form $\theta = iy$, $y > 0$, have corresponding roots $\theta = -iy$ also on the left boundary, and similarly roots on the right boundary of R of the form $\theta = \pi + iy$, $y > 0$, have corresponding roots $\theta = -\pi - iy + 2\pi = \pi - iy$, also on the right boundary. Since the cosine is also even and 2π-periodic, each of these corresponding pairs of roots of (20) yield identical eigenvalues of A through (14). We may thus focus entirely on finding

roots of (20) in the region R and may also exclude the portions of the boundary of R with $\mathrm{Im}(\theta) < 0$. Each such distinct root θ corresponds to a distinct eigenvalue λ.

From (18) and (8), the components of the eigenvectors are given by

$$v_j = v_1 q_j \left[g(j,\theta) + \frac{\alpha}{d} g(j-1,\theta) \right], \quad 1 < j \le n. \tag{23}$$

But, since $q_1 = 1$, $g(0,\theta) = 0$, and $g(1,\theta) = 1$, we may express the eigenvectors as

$$v_j = \begin{cases} q_j \left[\sin(j\theta) + \frac{\alpha}{d} \sin\big((j-1)\theta\big) \right] & \text{if } \theta \notin \{0,\pi\}, \\ q_j \left[j + \frac{\alpha}{d}(j-1) \right] & \text{if } \theta = 0, \\ q_j (-1)^{j-1} \left[j - \frac{\alpha}{d}(j-1) \right] & \text{if } \theta = \pi, \end{cases} \qquad 1 \le j \le n. \tag{24}$$

3. Special cases. In this section we examine various relationships between the matrix parameters d, α, and β which, when enforced, allow (20) to be solved explicitly or to be simplified to the form $F(\theta) = p(d,\alpha,\beta)$. These simplifications are the result of standard trigonometric identities. We list here several simplifications that we use for our function g:

$$g(j+1,\theta) + g(j-1,\theta) = 2g(j,\theta)\cos\theta, \tag{25}$$

$$g(j+1,\theta) - g(j-1,\theta) = 2\cos(j\theta), \tag{26}$$

$$g(j,\theta) + g(j-1,\theta) = g(2j-1,\theta/2), \tag{27}$$

$$g(j,\theta) - g(j-1,\theta) = \begin{cases} \cos\big((2j-1)\theta/2\big)/\cos(\theta/2) & \text{if } 0 < \theta < \pi, \\ 1 & \text{if } \theta = 0, \\ (-1)^{j-1}(2j-1) & \text{if } \theta = \pi. \end{cases} \tag{28}$$

Note that all of the above are identities for g and are valid for all $\theta \in \mathbb{C}$, including $\theta = k\pi$, $k \in \mathbb{Z}$.

3.1. Explicit solutions.

3.1.1. $\alpha = \beta = 0$. If $\alpha = \beta = 0$, then the matrix A, although not Toeplitz, has the same eigenvalues as the corresponding Toeplitz matrix ($a_i = a$ and $c_i = c$ for all i), since (20) collapses to $g(n+1,\theta) = 0$, whose solutions are $\theta_m = \frac{m\pi}{n+1}$, $1 \le m \le n$, giving eigenvalues

$$\lambda_m = b + 2d\cos\left(\frac{m\pi}{n+1}\right), \qquad 1 \le m \le n, \tag{29}$$

with corresponding eigenvectors

$$v_j^{(m)} = q_j \sin\left(\frac{jm\pi}{n+1}\right), \qquad 1 \le j \le n,\ 1 \le m \le n. \tag{30}$$

3.1.2. $\alpha = 0$, $\beta = d$. If $\alpha = 0$, $\beta = d$, then by using (27), (20) becomes

$$g(2n+1,\theta/2) = 0,$$

whose solutions are $\theta_m = \frac{2m\pi}{2n+1}$, giving eigenvalues

$$\lambda_m = b + 2d\cos\left(\frac{2m\pi}{2n+1}\right), \qquad 1 \le m \le n, \tag{31}$$

and corresponding eigenvectors

$$v_j^{(m)} = q_j \sin\left(\frac{2mj\pi}{2n+1}\right), \qquad 1 \le j \le n,\ 1 \le m \le n. \tag{32}$$

3.1.3. $\alpha = d$, $\beta = 0$. If $\alpha = d$, $\beta = 0$, the eigenvalues are the same as the above case (31), and the eigenvectors are

$$v_j^{(m)} = q_j \cos\left(\frac{(2j-1)2m\pi}{2(2n+1)}\right), \qquad 1 \le j \le n,\ 1 \le m \le n. \tag{33}$$

3.1.4. $\alpha = 0$, $\beta = -d$. If $\alpha = 0$, $\beta = -d$, we may use (28) to reduce (20) to

$$\cos\big((2n+1)\theta/2\big) = 0,$$

whose solutions are $\theta_m = (2m-1)\pi/(2n+1)$, $1 \le m \le n$, giving eigenvalues

$$\lambda_m = b + 2d\cos\left(\frac{(2m-1)\pi}{2n+1}\right), \qquad 1 \le m \le n, \tag{34}$$

and corresponding eigenvectors

$$v_j^{(m)} = q_j \sin\left(\frac{j(2m-1)\pi}{2n+1}\right), \qquad 1 \le j \le n,\ 1 \le m \le n. \tag{35}$$

3.1.5. $\alpha = -d$, $\beta = 0$. If $\alpha = -d$, $\beta = 0$, the eigenvalues are the same as the previous case (34), and the corresponding eigenvectors are

$$v_j^{(m)} = q_j \cos\left(\frac{(2j-1)(2m-1)\pi}{2(2n+1)}\right), \qquad 1 \le j \le n,\ 1 \le m \le n. \tag{36}$$

3.1.6. $\alpha = d$, $\beta = -d$. If $\alpha = -\beta = d$, (20) simplifies by using (26) to become $\cos(n\theta) = 0$. Hence $\theta_m = \frac{(2m-1)\pi}{2n}$, $1 \le m \le n$, and the eigenvalues are

$$\lambda_m = b + 2d\cos\left(\frac{(2m-1)\pi}{2n}\right), \qquad 1 \le m \le n, \tag{37}$$

with corresponding eigenvectors

$$v_j^{(m)} = q_j \sin\left(\frac{(2j-1)(2m-1)\pi}{4n}\right), \qquad 1 \le j \le n,\ 1 \le m \le n. \tag{38}$$

3.1.7. $\alpha = -d$, $\beta = d$. If $\alpha = -\beta = -d$, the eigenvalues are the same as the previous case (37), and the corresponding eigenvectors are

$$v_j^{(m)} = q_j \cos\left(\frac{(2j-1)(2m-1)\pi}{4n}\right), \qquad 1 \le j \le n,\ 1 \le m \le n. \tag{39}$$

3.1.8. $\alpha\beta = d^2$. Now consider the case $\alpha\beta = d^2$. Whereas the previous special cases were two-dimensional real manifolds in (d, α, β)-space, this case is a four-dimensional manifold in $\mathbb{R}^6$. By using (25), (20) becomes

$$g(n,\theta)\left(2\cos\theta + \frac{\alpha+\beta}{d}\right) = 0.$$

From this we immediately obtain

$$\theta_m = \frac{m\pi}{n}, \quad 1 \le m < n, \qquad \text{or} \qquad 2d\cos\theta_n = -(\alpha+\beta),$$

giving eigenvalues

$$\lambda_m = b + 2d\cos\left(\frac{m\pi}{n}\right), \quad 1 \le m < n, \qquad \text{and} \qquad \lambda_n = b - (\alpha + \beta). \tag{40}$$

The more specialized cases with $\alpha = \beta = d$ or $\alpha = \beta = -d$ were the only ones with $\alpha\beta = d^2$ that were considered by Yueh. In the case $\alpha = \beta = d$, by using (24) and (27), the eigenvectors are

$$v_j^{(m)} = q_j \sin\left(\frac{(2j-1)m\pi}{2n}\right), \qquad 1 \le j \le n,\ 1 \le m < n, \tag{41}$$

and, for the nth eigenpair, $\theta_n = \pi$, $\lambda_n = b - 2d$, and

$$v_j^{(n)} = q_j(-1)^{j-1}, \qquad 1 \le j \le n, \tag{42}$$

which we note is the same form as (41) with $m = n$. The case $\alpha = \beta = -d$ gives eigenvectors

$$v_j^{(m)} = q_j \cos\left(\frac{(2j-1)m\pi}{2n}\right), \qquad 1 \le j \le n,\ 1 \le m < n, \tag{43}$$

and $\theta_n = 0$, $\lambda_n = b + 2d$, and, by using (28),

$$v_j^{(n)} = q_j, \qquad 1 \le j \le n. \tag{44}$$

Outside these more specialized cases, it is impossible for θ_n to be zero or π, and so, from (24), we may write the eigenvectors as

$$v_j^{(m)} = q_j \left[\sin\left(\frac{jm\pi}{n}\right) + \frac{\alpha}{d}\sin\left(\frac{(j-1)m\pi}{n}\right)\right], \qquad 1 \le j \le n,\ 1 \le m < n, \tag{45}$$

and

$$v_j^{(n)} = q_j \left[\sin(j\theta_n) + \frac{\alpha}{d}\sin\big((j-1)\theta_n\big)\right], \qquad 1 \le j \le n, \tag{46}$$

where $\theta_n = \arccos(-(\alpha + \beta)/2d)$.

3.2. Eigenvalue distribution results. Here we examine various three- and four-dimensional real manifolds in (d, α, β)-space, where information on the distribution of the eigenvalues can be inferred. For this analysis it is convenient to partition the interval $[0, \pi]$ into subintervals in one of several ways. In the first way, $[0, \pi]$ is partitioned into $n+1$ subintervals, the first and last of which have width $1/(2n)$ while the remaining have width $1/n$:

$$\begin{aligned} I_0 &= \left[0, \frac{\pi}{2n}\right), \\ I_k &= \left[\frac{(2k-1)\pi}{2n}, \frac{(2k+1)\pi}{2n}\right), \qquad 1 \le k < n-1, \\ I_n &= \left[\frac{(2n-1)\pi}{2n}, \pi\right]. \end{aligned} \tag{47}$$

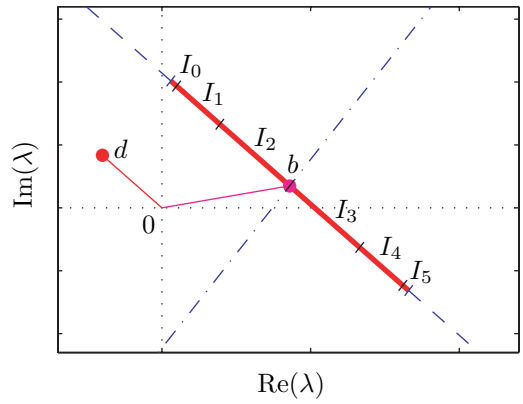


FIG. 1. *The location of eigenvalues $\lambda = b + 2d\cos(\theta)$ in the complex plane for various θ. If $\theta \in [0, \pi]$, then λ lies on the thick solid line segment. For illustration, this segment is divided into the images of the intervals I_k, $0 \le k \le n$, $n = 5$. Other partitionings of $[0, \pi]$ would divide the line segment differently. The dashed line extending beyond this line segment shows the location of λ when $\theta = iy$ or $\theta = \pi + iy$, $y \in \mathbb{R}$. The dotted-dashed line orthogonal to the thick line segment is the image of the line $\theta = \pi/2 \pm iy$, $y \in \mathbb{R}$.*

A second useful partitioning of $[0, \pi]$ is one where the subintervals have width $2/(2n+1)$ except for the first, which has half that width:

$$
\begin{aligned}
J_0 &= \left[0, \frac{\pi}{2n+1}\right), \\
J_k &= \left[\frac{(2k-1)\pi}{2n+1}, \frac{(2k+1)\pi}{2n+1}\right), \qquad 1 \le k < n-1, \\
J_n &= \left[\frac{(2n-1)\pi}{2n+1}, \pi\right].
\end{aligned} \tag{48}
$$

Finally, consider partitioning $[0, \pi]$ into n equal length intervals:

$$
\begin{aligned}
K_k &= \left[\frac{(k-1)\pi}{n}, \frac{k\pi}{n}\right), \qquad 1 \le k < n, \\
K_n &= \left[\frac{(n-1)\pi}{n}, \pi\right].
\end{aligned} \tag{49}
$$

Typically, the following results make statements such as "Under the stated assumptions, there is one real solution of (20) lying in each of the intervals I_0 to I_{n-1}." The eigenvalues of A are given by $\lambda = b + 2d\cos(\theta)$. Real values of θ on the interval $[0, \pi]$ thus correspond to eigenvalues on a line segment of length $4|d|$ parallel to the ray from the origin to d and centered at b, as depicted in Figure 1. The various subintervals defined above correspond to portions of this line segment. Sometimes complex solutions for θ of the form $\theta = iy$, $\theta = \pi + iy$, or $\theta = \pi/2 \pm iy$, $y \in \mathbb{R}$, are also shown to exist. In the first two cases, the corresponding eigenvalues are $\lambda = b \pm 2d\cosh(y)$, respectively, which lie on the same line as the intervals shown in Figure 1 but further away from b. In the case $\theta = \pi/2 \pm iy$, the corresponding eigenvalues are $\lambda = b \pm i2d\sinh(y)$, which lie an equal distance from b on a line through b orthogonal to the ray through d. Of course, if b and d are real, then each of these distinct real solutions for θ and the complex solutions with $\mathrm{Re}(\theta) \in \{0, \pi\}$ yield distinct real eigenvalues for A.

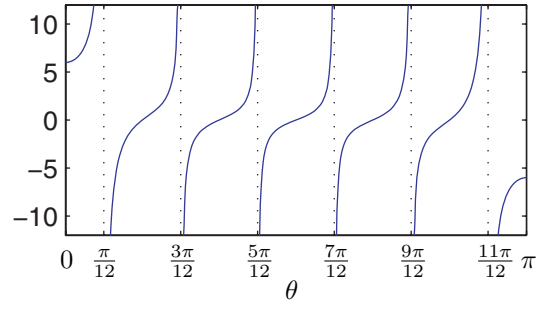


FIG. 2. *The function* $\theta \mapsto \tan(n\theta)/\sin(\theta)$, $\theta \in [0, \pi]$, *for the case* $n = 6$. *The intervals* I_k, $0 \le k \le n$, *are demarcated by the dotted vertical lines.*

For some of this analysis it is convenient to express the necessary and sufficient condition for θ, (20), in an equivalent form. By expanding $\sin((n \pm 1)\theta)$ and simplifying, (20) becomes

$$\frac{\sin(n\theta)}{\sin(\theta)} \left[\left(1 + \frac{\alpha\beta}{d^2} \right) \cos(\theta) + \frac{\alpha + \beta}{d} \right] + \cos(n\theta) \left(1 - \frac{\alpha\beta}{d^2} \right) = 0. \tag{50}$$

3.2.1. $\boldsymbol{\alpha\beta = -d^2}$**.** Consider the four-dimensional manifold $\alpha\beta = -d^2$, but exclude the cases $\alpha = -\beta = \pm d$ as these have already been considered. In this case, (50) simplifies to

$$\frac{\sin(n\theta)}{\sin(\theta)} \left[\frac{\alpha + \beta}{d} \right] + 2\cos(n\theta) = 0. \tag{51}$$

Since we have excluded the possibility that $\alpha + \beta = 0$, for (51) to hold $\cos(n\theta)$ cannot be zero, and hence we may write this equation as

$$\frac{\tan(n\theta)}{\sin(\theta)} = \frac{-2d}{\alpha + \beta}. \tag{52}$$

Let $F(\theta) = \tan(n\theta)/\sin(\theta)$ and $p(d, \alpha, \beta) = -2d/(\alpha + \beta)$, and assume that the value of p is real. (We are now restricted to a three-dimensional manifold.) The function F on $[0, \pi]$ is an odd function with respect to $\pi/2$. It monotonically increases from $-\infty$ to $+\infty$ on each interval I_k, $1 \le k < n$. On the interval I_0 it monotonically increases from n to $+\infty$, and on the interval I_n it monotonically increases from $-\infty$ to $-n$; see Figure 2. Consequently, if $|p| \ge n$, there are exactly n distinct real solutions of (52) for θ in $[0, \pi]$, one in each of the intervals $I_1, \ldots, I_{n-1}$ and the last one in either I_0 (if $p > 0$) or in I_n (if $p < 0$). These n distinct values for θ correspond to n distinct real eigenvalues λ. If $|p| < n$, then there are only $n - 1$ real solutions of (52), one in each of the intervals $I_1, \ldots, I_{n-1}$, corresponding to $n - 1$ real eigenvalues for A. In the event that all variables in A are real, the last eigenvalue must also be real, since they must occur in complex conjugate pairs. (It turns out that, even if just b, d, and p are real, the last eigenvalue is also real.) But for $\theta = x + iy$, $x, y \in \mathbb{R}$,

$$\cos(\theta) = \cos(x + iy) = \cosh(y)\cos(x) - i\sinh(y)\sin(x),$$

thus $\lambda = b + 2d\cos(\theta)$ will be real if b and d are real, and θ is real or $\mathrm{Re}(\theta) = k\pi$. We are thus led to look for the nth solution of (52) on the boundary of R by taking

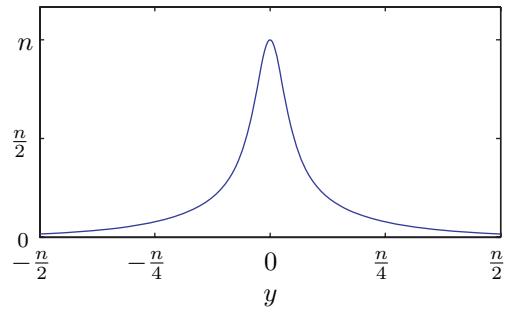


FIG. 3. *The function* $y \mapsto \tanh(ny)/\sinh(y)$.

$\theta = iy$ or $\theta = \pi + iy$, $y > 0$. (Recall that on the boundary of R we need only consider $y \geq 0$.) If $0 < p < n$, substituting $\theta = iy$, $y \in \mathbb{R}$, into (52) gives

$$\frac{\tanh(ny)}{\sinh(y)} = p. \tag{53}$$

A plot of this function is shown in Figure 3. Since it monotonically decreases from n to zero as y increases from zero, there is exactly one solution y^* of (53), with $y^* > 0$. If $-n < p < 0$, substituting $\theta = \pi + iy$, $y \in \mathbb{R}$, into (52) gives

$$\frac{\tanh(ny)}{\sinh(y)} = -p, \tag{54}$$

again yielding exactly one more root y^*, with $y^* > 0$. If $|p|$ is large ($|\alpha + \beta|$ is small), then the solutions for θ asymptotically approach $\theta_m = (2m-1)\pi/(2n)$, $1 \leq m \leq n$, as expected since these are the solutions for $\alpha = -\beta = \pm d$. But in any event the real solutions of (52) are approximately spaced by a distance of π/n as is readily apparent in Figure 2.

3.2.2. $\alpha\beta = 0$. If the product $\alpha\beta$ is zero (but α and β are not both zero and one is not $\pm d$, as these cases have been previously considered), then (20) becomes

$$g(n+1,\theta) + \frac{\alpha+\beta}{d} g(n,\theta) = 0. \tag{55}$$

This can be written as

$$\frac{\sin\big((n+1/2)\theta + \theta/2\big)}{\sin(2\theta/2)} + \left(\frac{\alpha+\beta}{d}\right) \frac{\sin\big((n+1/2)\theta - \theta/2\big)}{\sin(2\theta/2)} = 0,$$

which upon expanding and simplifying yields

$$\left(1 + \frac{\alpha+\beta}{d}\right)\left[\frac{\sin\big((2n+1)\theta/2\big)}{\sin(\theta/2)}\right] + \left(1 - \frac{\alpha+\beta}{d}\right)\left[\frac{\cos\big((2n+1)\theta/2\big)}{\cos(\theta/2)}\right] = 0.$$

Now, since we are assuming that $\alpha + \beta \neq \pm d$, both factors in parentheses in the above expression are nonzero. This means that if one of the factors in square brackets is zero, both must be zero for the equation to hold. However, this is impossible since the first factor in square brackets is $g(2n+1, \theta/2)$, which is zero only when $(2n+1)\theta/2 = k\pi$,

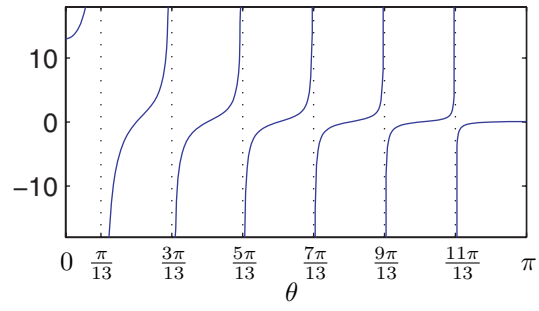


FIG. 4. *The function* $\theta \mapsto \tan((2n+1)\theta/2)/\tan(\theta/2)$, $\theta \in [0,\pi]$, *for the case* $n = 6$. *The intervals* J_k, $0 \le k \le n$, *are demarcated by the dotted vertical lines.*

$1 \le k \le n$, which implies that $\cos((2n+1)\theta/2) = \cos(k\pi) \neq 0$. We may thus validly rearrange the above expression to obtain

$$\frac{\tan\big((2n+1)\theta/2\big)}{\tan(\theta/2)} = \frac{\alpha+\beta-d}{\alpha+\beta+d}. \tag{56}$$

Let $F(\theta)$ and $p(d,\alpha,\beta)$ be the left and right sides of the above equation, respectively, and assume that the value of p is real. (Again, we are now restricted to a three-dimensional manifold.) The function F on $[0,\pi]$ is plotted in Figure 4 for the case $n = 6$. F is monotonically increasing from $-\infty$ to $+\infty$ on the intervals J_k, $1 \le k < n$, monotonically increasing from $2n+1$ to $+\infty$ on J_0, and monotonically increasing from $-\infty$ to $1/(2n+1)$ on J_n. Thus if p is real and at least as big as $2n+1$, there are exactly n real solutions, one in each of the intervals J_k, $0 \le k \le n-1$. If p is real and equal to or smaller than $1/(2n+1)$, there is one real solution in each of the intervals J_k, $1 \le k \le n$. All of these real roots are approximately separated by a distance of $2\pi/(2n+1)$. Substituting $\theta = iy$, $y \in \mathbb{R}$, into (56) gives

$$\frac{\tanh\big((2n+1)y/2\big)}{\tanh(y/2)} = p, \tag{57}$$

while substituting $\theta = \pi + iy$, $y \in \mathbb{R}$, into (56) gives

$$\frac{\coth\big((2n+1)y/2\big)}{\coth(y/2)} = p. \tag{58}$$

The functions on the left sides of the above expressions are shown in Figure 5. The first decreases monotonically from $2n+1$ to 1, while the second increases from $1/(2n+1)$ to 1 as y increases from zero. Thus if $1/(2n+1) < p < 1$, there is one solution of (56) of the form $\theta = \pi + iy^*$, $y^* > 0$, and if $1 < p < (2n+1)$, there is one solution of the form $\theta = iy^*$, $y^* > 0$. (Note that since $d \neq 0$ it is impossible for $p = 1$.)

3.2.3. $\alpha + \beta = 0$. If $\alpha + \beta = 0$, then (50) may be written as

$$\left(1 + \frac{\alpha\beta}{d^2}\right)\left[\frac{\sin(n\theta)\cos(\theta)}{\sin(\theta)}\right] + \left(1 - \frac{\alpha\beta}{d^2}\right)\cos(n\theta) = 0. \tag{59}$$

The cases $\alpha\beta = 0$ and $\alpha\beta = \pm d^2$ have already been considered, so here we may assume that neither factor in parentheses is zero and these two factors are not equal.

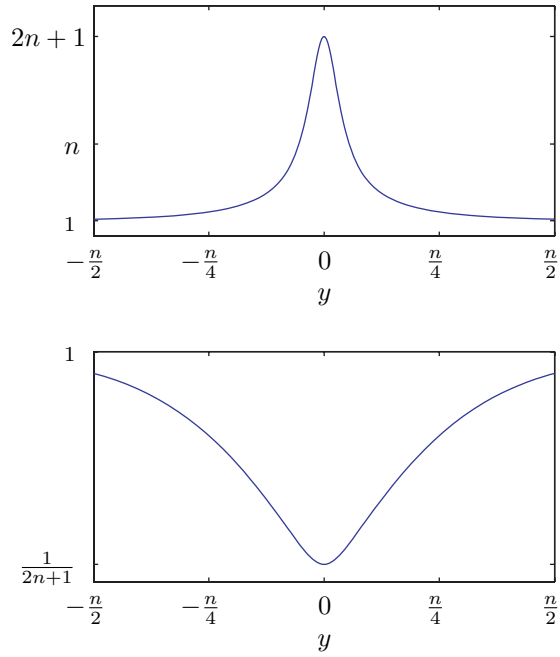


FIG. 5. *Top: The function* $y \mapsto \tanh((2n+1)y/2)/\tanh(y/2)$. *Bottom: The function* $y \mapsto \coth((2n+1)y/2)/\coth(y/2)$.

If $\cos(n\theta) = 0$, then $\theta_m = \frac{(2m-1)\pi}{2n}$, $1 \le m \le n$, are all of the possible values of θ in $[0, \pi]$. The factor $\sin(n\theta)$ clearly cannot be zero for these θ_m; hence, for the equation to hold, $\cos\theta_m$ must be zero, that is, $\theta_m = \pi/2$, which means that n must be odd and $m = (n+1)/2$. Thus the condition on θ may be written

$$\frac{\tan(n\theta)}{\tan(\theta)} = \frac{\alpha\beta - d^2}{\alpha\beta + d^2}, \qquad \theta \neq \pi/2, \tag{60}$$

where $\theta = \pi/2$ is an additional solution in the case n is odd. (Thus one eigenvalue of A is always b when n is odd and $\alpha + \beta = 0$.) Again, let $F(\theta)$ and $p(d, \alpha, \beta)$ be the left and right sides of (60), respectively, and suppose that the value of p is real. Figure 6 displays a graph of F on $[0, \pi]$ for both an even and an odd n case. F is even around $\pi/2$ and is equal to n at both $\theta = 0$ and $\theta = \pi$. The subintervals of interest are the I_k given by (47). On I_0 the function F increases monotonically from n to $+\infty$, and on intervals I_k, $1 \le k \le n/2 - 1$, it increases monotonically from $-\infty$ to $+\infty$. On the center interval(s) ($I_{n/2}$ if n is even, $I_{(n-1)/2} \cup I_{(n+1)/2}$ if n is odd), F increases from $-\infty$ to a maximum value of zero at $\pi/2$ and then decreases back to $-\infty$. The behavior on the other intervals is dictated by the fact that F is even around $\pi/2$. Thus if $p \ge n$, there is one real solution of (60) on each of the intervals I_k, $0 \le k \le n/2 - 1$, and $n/2 + 1 \le k \le n$. If p is smaller than zero, there is one real solution of (60) on each of the intervals I_k, $1 \le k \le n/2 - 1$, and $n/2 + 1 \le k \le n - 1$ and two in the center interval(s) ($I_{n/2}$ if n is even, $I_{(n-1)/2} \cup I_{(n+1)/2}$ if n is odd). This accounts for all n solutions for θ (provided we add in the additional solution $\theta = \pi/2$ if n is odd). As $|p|$ gets large ($\alpha\beta \to -d^2$), these solutions approach $\theta_m = (2m-1)\pi/(2n)$, as expected.

Now suppose that $0 \le p < n$. First, if $1 < p < n$, there is one real solution of (60) on each of the intervals I_k, $1 \le k < n/2 - 1$, and $n/2 + 1 \le k \le n - 1$. By adding in

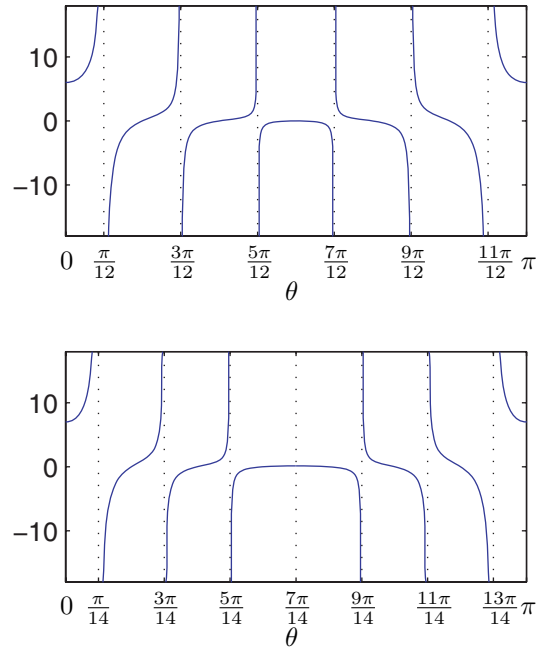


FIG. 6. *The function* $\theta \mapsto \tan(n\theta)/\tan(\theta)$, $\theta \in [0, \pi]$, *for an even case,* $n = 6$ *(top), and an odd case,* $n = 7$ *(bottom). The intervals* I_k, $0 \le k \le n$, *are demarcated by the dotted vertical lines.*

the solution $\theta = \pi/2$ if n is odd, this accounts for $n-2$ real solutions for θ. However, a pair of complex solutions $\theta = iy$ and $\theta = \pi + iy$ exist, since both of these values for θ, when substituted into (60), yield

$$\frac{\tanh(ny)}{\tanh(y)} = p, \tag{61}$$

which has a unique solution y^*, with $y^* > 0$, when $1 < p < n$. (The plot of $\tanh(ny)/\tanh(y)$ is similar to that depicted in the top panel of Figure 5 but with maximum value n rather than $2n+1$.) It is impossible to have $d \neq 0$ and $p = 1$, and, since we have dealt with $\alpha\beta = d^2$ previously, we have assumed that $p \neq 0$; however, we have not yet dealt with the case $0 < p < 1$. As p increases through zero, two real roots annihilate each other at $\pi/2$, and we therefore substitute $\theta = \pi/2 + iy$ into (60) yielding

$$\frac{\tanh(ny)}{\coth(y)} = p. \tag{62}$$

The function on the left side of (62) is plotted in Figure 7, where we note that there are two solutions $\pm y^*$ for each value of $p \in (0, 1)$. These two solutions for y correspond to two distinct values θ in R and a pair of eigenvalues λ given by

$$\lambda = b \pm i2d\sinh(y^*).$$

If b and d are real, these are a complex conjugate pair.

3.2.4. $(\alpha + \beta)/d$ and $\alpha\beta/d^2$ are real. If the matrix parameters d, α, and β are such that the quantities $(\alpha + \beta)/d$ and $\alpha\beta/d^2$ are real, then we can conclude

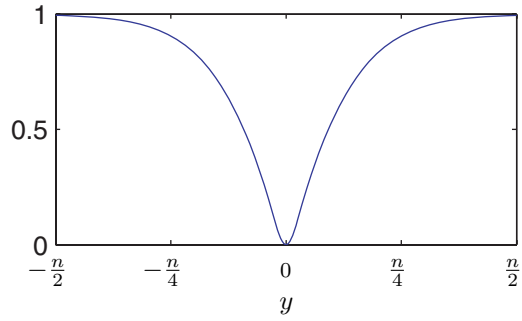


FIG. 7. *The function* $y \mapsto \tanh(ny)/\coth(y)$.

that there are at least $n-2$ real solutions to (20) and provide sufficient conditions under which n real solutions exist. We shall assume that $\alpha\beta \neq d^2$ since that case was previously discussed.

The quantities $(\alpha+\beta)/d$ and $\alpha\beta/d^2$ are both real if and only if either both α/d and β/d are real or they are complex conjugates, which we may express as one of the following two options:

1. $\alpha = sd$, $\beta = td$, $s, t \in \mathbb{R}$,
2. $\alpha = (s+it)d$, $\beta = (s-it)d$, $s, t \in \mathbb{R}$.

Thus the manifold of points that we are considering is four-dimensional. Note that we have not restricted ourselves to the situation where the matrix is real, although that case is included as part of option 1 above.

Denote the left side of (20) (or equivalently the left side of (50)) as $T(\theta)$, so that the necessary and sufficient condition for θ is $T(\theta) = 0$. From (20) and (16) we have

$$\begin{aligned} T(0) &= (n+1) + \frac{\alpha+\beta}{d}n + \frac{\alpha\beta}{d^2}(n-1) \\ &= \left(1 + \frac{\alpha+\beta}{d} + \frac{\alpha\beta}{d^2}\right)n + \left(1 - \frac{\alpha\beta}{d^2}\right) \end{aligned}$$

and

$$\begin{aligned} T(\pi) &= (1-)^n(n+1) + (-1)^{n-1}\frac{\alpha+\beta}{d}n + (-1)^{n-2}\frac{\alpha\beta}{d^2}(n-1) \\ &= (-1)^n\left[\left(1 - \frac{\alpha+\beta}{d} + \frac{\alpha\beta}{d^2}\right)n + \left(1 - \frac{\alpha\beta}{d^2}\right)\right]. \end{aligned}$$

Evaluating T at $\theta = m\pi/n$, $1 \leq m < n$, via (50) immediately gives

$$T\left(\frac{m\pi}{n}\right) = (-1)^m\left(1 - \frac{\alpha\beta}{d^2}\right).$$

Since we are assuming that $\alpha\beta \neq d^2$, the continuous function T alternates sign at the points $m\pi/n$, $1 \leq m < n$. This immediately implies that there are $n-2$ real roots of T, one in each of the intervals $K_2, K_3, \ldots, K_{n-1}$. The interval K_1 will contain an additional root provided $T(0)$ is zero or has opposite sign from $T(\pi/n)$, that is, $T(0)/T(\pi/n) \leq 0$. This can be expressed as

$$\frac{1 + \frac{\alpha+\beta}{d} + \frac{\alpha\beta}{d^2}}{1 - \frac{\alpha\beta}{d^2}} \geq -\frac{1}{n}. \tag{63}$$

Analogously, the interval K_n will contain an additional root provided

$$\frac{1-\frac{\alpha+\beta}{d}+\frac{\alpha\beta}{d^2}}{1-\frac{\alpha\beta}{d^2}} \geq -\frac{1}{n}. \tag{64}$$

The inequalities (63) and (64) are sufficient conditions for a root to exist in K_1 or K_n, respectively. However, it is possible that two roots exist in one of these intervals if both inequalities fail to hold.

Substituting $\theta = iy$ or $\theta = \pi + iy$, $y \in \mathbb{R}$, into (50) yields the equivalent condition

$$\frac{\tanh(ny)}{\sinh(y)}\left[\frac{1+\frac{\alpha\beta}{d^2}}{1-\frac{\alpha\beta}{d^2}}\cosh(y) \pm \frac{\frac{\alpha+\beta}{d}}{1-\frac{\alpha\beta}{d^2}}\right] = -1, \tag{65}$$

where the + sign is for $\theta = iy$ and the − sign for $\theta = \pi + iy$. The function $y \to \tanh(ny)(X\cosh(y)+Z)/\sinh(y)$ has limiting values of $n(X+Z)$ and X as y tends to 0 and ∞, respectively. However, it is not necessarily monotonic on $y > 0$. Nonetheless, since it is continuous, we can conclude the following. If -1 lies between the values

$$n\left(\frac{1+\frac{\alpha+\beta}{d}+\frac{\alpha\beta}{d^2}}{1-\frac{\alpha\beta}{d^2}}\right) \qquad \text{and} \qquad \frac{1+\frac{\alpha\beta}{d^2}}{1-\frac{\alpha\beta}{d^2}},$$

there will be at least one solution of (65) in the form $\theta = iy$, $y \in \mathbb{R}$. If -1 lies between the values

$$n\left(\frac{1-\frac{\alpha+\beta}{d}+\frac{\alpha\beta}{d^2}}{1-\frac{\alpha\beta}{d^2}}\right) \qquad \text{and} \qquad \frac{1+\frac{\alpha\beta}{d^2}}{1-\frac{\alpha\beta}{d^2}},$$

there will be at least one solution of (65) in the form $\theta = \pi + iy$, $y \in \mathbb{R}$. (Note that either of the two values in each of the above pairs may be the smaller one, depending on d, α, and β.)

4. Conclusion. The eigenvalues of the tridiagonal matrix A given by (1) are of the form $\lambda_m = b+2d\cos(\theta_m)$, where θ_m are the solutions to the nonlinear equation (20) in the region R defined by (21). The corresponding eigenvectors are given by (24).

The space of matrix parameters d, α, and β is $\mathbb{C}^3$, which can be identified with $\mathbb{R}^6$. Restriction to a number of two-dimensional manifolds in this space permits explicit solutions of (20), and these cases were itemized. In addition, the four-dimensional manifold of points $\alpha\beta = d^2$ also yields explicit solutions, and these were given. Information about the distribution of eigenvalues was also described for several other three- and four-dimensional manifolds. Many of these were cases where (20) could be separated into the form $F(\theta) = p(d, \alpha, \beta)$, where F is a certain ratio of trigonometric functions of multiples of θ and p is real-valued. In most of these cases, this equation has either n real roots or $n-1$ real roots and one complex root, but the corresponding eigenvalues are all real (assuming that b and d are also real). In one case, $\alpha+\beta = 0$, a single complex conjugate pair of eigenvalues is possible if $0 < (\alpha\beta-d^2)/(\alpha\beta+d^2) < 1$. The four-dimensional manifold of points specified by the requirement that both the quantities $(\alpha+\beta)/d$ and $\alpha\beta/d^2$ be real was shown to yield at least $n-2$ real eigenvalues for A. Sufficient conditions for when the remaining two eigenvalues are also real were provided. This case includes the case where the matrix A is real to begin with.

Efficient numerical algorithms exist, based on the QR factorization, for the computation of the eigenvalues of general tridiagonal matrices [2]. The nonlinear equation (20) we have presented here would generally need to be solved with a root-finding

algorithm to find the θ_m in the region R, which may or may not be as efficient as the QR-based algorithms. However, the various special cases that we have enumerated provide additional information on the distribution of the solutions θ_m, specifying various subintervals of $[0, \pi]$ wherein exactly one solution must lie. Within these subintervals, more refined estimates of the locations of solutions are often easily obtained by plotting the appropriate function F described herein and noting where it crosses the value of p.

Tridiagonal matrices frequently occur in applications where it is desirable to know their eigenvalues. As just one example, consider a simple Markov process with n states arranged in a chain formation where $x_i(t)$ is the probability of being in state i at time t, $i = 1, \ldots, n$. Let a be the rate at which a particle moves to the right (from state i to state $i+1$) and c the rate at which a particle moves to the left through the states:

$$x_1 \underset{c}{\overset{a}{\longleftrightarrow}} x_1 \underset{c}{\overset{a}{\longleftrightarrow}} x_2 \cdots x_{n-1} \underset{c}{\overset{a}{\longleftrightarrow}} x_n. \tag{66}$$

Such systems are commonly employed as models or parts of models for ion channel gating [1]. The governing system is

$$x' = Ax, \tag{67}$$

where A is given by (1) with $b = -a - c$, $d = \sqrt{ac}$, $\alpha = -c$, and $\beta = -a$. From (40) we immediately conclude that the eigenvalues are

$$\lambda_m = -(a + c) + 2\sqrt{ac}\cos\left(\frac{m\pi}{n}\right), \quad 1 \le m < n, \qquad \lambda_n = 0.$$

From (7), (45), and (46), the corresponding eigenvectors are

$$v_j^{(m)} = \left(\frac{a}{c}\right)^{(j-1)/2}\left[\sin\left(\frac{jm\pi}{n}\right) - \sqrt{\frac{c}{a}}\sin\left(\frac{(j-1)m\pi}{n}\right)\right], \quad 1 \le j \le n,\ 1 \le m < n,$$

and

$$v_j^{(n)} = \left(\frac{a}{c}\right)^{(j-1)/2}\left[\sin(j\theta_n) - \sqrt{\frac{c}{a}}\sin\big((j-1)\theta_n\big)\right], \qquad 1 \le j \le n,$$

where $\theta_n = \arccos\left((a + c)/2\sqrt{ac}\right)$.

REFERENCES

[1] F. Bezanilla, *The voltage sensor in voltage-dependent ion channels*, Physiol. Rev., 80 (2000), pp. 555–592.

[2] R. L. Burden and J. D. Faires, *Numerical Analysis*, 7th ed., Brooks/Cole, Forest Grove, CA, 2001.

[3] S. S. Cheng, *Partial Difference Equations*, CRC Press, New York, 2003.

[4] C. M. da Fonseca, *The characteristic polynomial of some perturbed tridiagonal k-Toeplitz matrices*, Appl. Math. Sci., 1 (2007), pp. 59–67.

[5] C. M. da Fonseca, *On the eigenvalues of some tridiagonal matrices*, J. Comput. Appl. Math., 200 (2007), pp. 283–286.

[6] S. Kouachi, *Eigenvalues and eigenvectors of tridiagonal matrices*, Electron. J. Linear Algebra, 15 (2006), pp. 115–133.

[7] W. Ledermann and G.E.H. Reuter, *Spectral theory for the differential equations of simple birth and death processes*, Philos. Trans. R. Soc. Lond. Ser. A, 246 (1954), pp. 321–369.

[8] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, PA, 2000.

[9] W.-C. Yueh, *Eigenvalues of several tridiagonal matrices*, Appl. Math. E-Notes, 5 (2005), pp. 66–74.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 657–683

ACCELERATION TECHNIQUES FOR APPROXIMATING THE MATRIX EXPONENTIAL OPERATOR*

M. POPOLIZIO† AND V. SIMONCINI‡

Abstract. In this paper we investigate some well-established and more recent methods that aim at approximating the vector $\exp(A)v$ when A is a large symmetric negative semidefinite matrix, by efficiently combining subspace projections and spectral transformations. We show that some recently developed acceleration procedures may be restated as preconditioning techniques for the partial fraction expansion form of an approximating rational function. These new results allow us to devise a priori strategies to select the associated acceleration parameters; theoretical and numerical results are shown to justify these choices. Moreover, we provide a performance evaluation among several numerical approaches to approximate the action of the exponential of large matrices. Our numerical experiments provide a new, and in some cases, unexpected picture of the actual behavior of the discussed methods.

Key words. Krylov subspace, matrix exponential, rational functions, iterative methods, large matrices

AMS subject classifications. 65F30, 65F10, 65F50

DOI. 10.1137/060672856

1. Introduction.

In this paper we are interested in the numerical approximation of the action of the matrix exponential to a vector, namely

$$y = \exp(A)v,$$

when the real $n \times n$ matrix A is large and symmetric negative semidefinite. In the following we assume that $\|v\| = 1$, where $\|\cdot\|$ denotes the Euclidean norm. We investigate some well-established and more recent methods that aim at approximating the vector y by efficiently combining subspace projections and spectral transformations. We refer the reader to [29] for a more complete recent survey.

Two apparently distinct classes of approaches have been discussed in literature when A is large and sparse. In the first type of strategy, the matrix is projected onto a possibly much smaller space, the exponential is then applied to the reduced matrix, and finally the approximation is projected back to the original large space. If H and e denote the projected and restricted versions of A and v, respectively, then this process can be summarized as follows:

$$y \approx V \exp(H)e,$$

where the columns of V form a basis of the projection space; see, e.g., [14], [15], [16], [23], [31], [32], [33], [34], [41]. In particular, van den Eshof and Hochbruck have recently devised an acceleration method based on a spectral transformation, which appears to significantly reduce the dimension of the approximation space without sacrificing accuracy and efficiency [41].

*Received by the editors October 20, 2006; accepted for publication (in revised form) by A. Frommer February 11, 2008; published electronically June 13, 2008.
http://www.siam.org/journals/simax/30-2/67285.html

†Dipartimento di Matematica, Università di Bari, Via E. Orabona 4, I-70125 Bari, Italy (popolizio@dm.uniba.it).

‡Dipartimento di Matematica, Università di Bologna, Piazza di Porta S. Donato, 5, I-40127 Bologna, Italy, and CIRSA, Ravenna, Italy (valeria@dm.unibo.it).

In the second class of methods, the exponential function is first approximated by an appropriate simpler function, and then the action of this matrix function is evaluated; see, e.g., [4], [8], [10], [27], [29], [31]. To this end, a special role is played by rational function approximation to the exponential; see, e.g., [3], [9], [11], [40]. Let $\mathcal{R}_\nu$ be such a rational function, so that $\mathcal{R}_\nu(A)v \approx \exp(A)v$, and assume that $\mathcal{R}_\nu$ admits the following partial fraction expansion:

$$\exp(A)v \;\approx\; \mathcal{R}_\nu(A)v = \omega_0 v + \sum_{j=1}^{\nu} \omega_j (A - \xi_j I)^{-1} v. \tag{1.1}$$

In this approach, an approximation to y can be obtained by first solving the shifted linear systems appearing in the sum, and then by collecting the sum terms; see, e.g., [4], [20]. The computation of the system solutions can be either carried out by a direct method, or, if memory requirements become unacceptable, by iterative methods such as Krylov subspace solvers [35]. In particular, some of these methods can fully exploit the recent developments in iterative linear system solvers; see, e.g., [39].

In this paper we argue that the distinction between the two aforementioned categories is in fact very vague, and that increased understanding can be gained by exploiting both viewpoints.

The aim of this paper is twofold. On the one hand, we show that the acceleration method by van den Eshof and Hochbruck cited above may be restated as a preconditioning technique of the rational function approximation in (1.1) when the exponential is replaced by an approximating rational function. In addition, we show that another recently proposed method (see [1]) may also be viewed as a preconditioning technique for appropriately solving the shifted systems in (1.1). These new results allow us to devise a priori strategies to select the associated acceleration parameters; our completely algebraic analysis complements proposed selections based in some cases (cf. [41]) on the [illegible] approximation of the solution to analytic problems; numerical results are shown to justify our choices.

Available comparisons of different schemes in the two categories above are very limited; see, e.g., [37]. Our second aim is to provide a performance evaluation among several numerical approaches to approximate the exponential, therefore filling a gap in the recent literature. Our numerical experiments show that the ranking of the methods changes significantly depending on whether linear systems can be solved by a direct method. In particular, our numerical findings highlight the competitiveness of the simple partial fraction expansion form in (1.1) over newly developed acceleration procedures when appropriate iterative methods are used. On the other hand, in the case when direct methods are applicable, ad hoc and acceleration techniques, such as (shift-invert) Lanczos, are superior to the partial fraction expansion method.

This paper is organized as follows. Section 2 reviews the role of rational functions in the matrix exponential context and recalls the notation and basic facts associated with the approximation in the Krylov subspace. Section 3 discusses an acceleration method based on the shift-invert Lanczos (SI), while section 3.1 provides a theoretical justification of the parameter selection in the shift-invert step. Some theoretical and computational guidelines for the method are reported in sections 3.2 and 3.3, respectively. Section 4 discusses a second acceleration method, and a cheaper strategy is proposed to deal with the acceleration matrix. The associated parameter is analyzed in section 4.1, where a theoretical justification for its selection is provided; some implementation improvements are discussed in section 4.2. Section 5 and its subsections report on our numerical experience with all analyzed methods. Finally, section

6 presents further experiments with enhanced implementations of the discussed acceleration procedures, while section 7 summarizes our final conclusions.

Throughout this paper we assume that the spectrum of A, $\mathrm{spec}(A)$, is contained in the interval $[\alpha, 0]$, for some $\alpha < 0$. This is not a restrictive assumption. Indeed, if $\mathrm{spec}(A) \subset [\alpha, \beta]$, with $\beta < 0$, then the spectrum of $A_1 = A - \beta I$ is contained in $[\alpha - \beta, 0]$ and $\exp(A) = \exp(A_1)\exp(\beta)$. Therefore the behavior of $\exp(A)$ can be recovered from that of $\exp(A_1)$. As we shall see, standard procedures are particularly slow for large values of $\|A\| = |\alpha|$, and thus in the context of acceleration procedures, our main interest will be in large $\|A\|$. Throughout our analysis we assume to work in exact arithmetic and we refer to the paper by Druskin, Greenbaum, and Knizhnerman [12] for an analysis of Krylov subspace methods for matrix functions in finite precision arithmetic.

2. Rational function approximation and Krylov subspaces. Let $\mathcal{R}_\nu(z) = \mathcal{N}_\nu(z)/\mathcal{D}_\nu(z)$ be a rational function approximating the exponential function, with $\mathcal{N}_\nu, \mathcal{D}_\nu$ polynomials of degree ν. When $\mathcal{R}_\nu$ is the rational Chebyshev function it holds (see [9] and references therein) that

$$\sup_{\lambda \geq 0} |\exp(-\lambda) - R_\nu(\lambda)| \approx 10^{-\nu},$$

which implies a similar estimate for $\|\exp(A)v - R_\nu(-A)\|$ when A is symmetric and negative semidefinite. Due to this nice approximation property, Chebyshev rational functions are commonly employed to approximate $\exp(A)v$ when A has a wide spectrum.

Let (1.1) be the partial fraction expansion of $\mathcal{R}_\nu$; note that Chebyshev rational functions have distinct poles, so that (1.1) can be correctly employed in this case. Since A is real, the poles in (1.1) come in complex conjugates; therefore we can write[1]

$$\mathcal{R}_\nu(A)v = \omega_0 v + \sum_{\substack{j=1 \\ j \text{ odd}}}^{\nu-1} 2\Re\left(\omega_j (A - \xi_j I)^{-1} v\right) + \omega_\nu (A - \xi_\nu I)^{-1} v, \tag{2.1}$$

where ξ_ν denotes the real pole if ν is odd.

When dealing with large dimension problems, the shifted systems can be solved by means of iterative methods. The simplified quasi-minimal residual (QMR) method [18] can be used to obtain an approximation to $x^{(j)} = (A - \xi_j I)^{-1} v$ separately for each j. The method is an appropriately refined version of the non-Hermitian Lanczos algorithm, which exploits the (shifted) complex symmetric form of the coefficient matrix to devise a single short-term recurrence. Preconditioning can be successfully applied to this variant as long as the preconditioner is also complex symmetric. We refer to this approach as PFE+QMR in our numerical experiments of section 5.2. An alternative choice is to use the single Krylov subspace $K_k(A, v) = \mathrm{span}\{v, Av, \ldots, A^{k-1}v\}$ as the approximation space. Assuming a zero initial solution approximation, for each j the kth iterate $x_k^{(j)}$ belongs to $K_k(A - \xi_j I, v) = K_k(A, v)$, where the last equality

[1] The use of Chebyshev functions implies a change of sign of the coefficient matrix in $\mathcal{R}_\nu(-A)$ and in its partial fraction expansion. If ζ denotes a Chebyshev pole, then the system to be solved is $(-A - \zeta I)x = v$, which is equivalent to $(A - \xi I)x = -v$, with $\xi = -\zeta$. In the following we omit specifying this change, and in practice, it is sufficient to change each pole's sign when Chebyshev approximation is used.

is due to the shift invariance of Krylov subspaces. Then the linear combination

$$x_k = \omega_0 v + \sum_{j=1}^{\nu} \omega_j x_k^{(j)} \in K_k(A, v) \tag{2.2}$$

is an approximation to $\mathcal{R}_\nu(A)v$. To speed up convergence without loosing the shifted form, it is possible to precondition all systems with the same matrix, say $(A - \tau I)^{-1}$ with $\tau > 0$, namely

$$(A - \tau I)^{-1}(A - \xi I)x = (A - \tau I)^{-1}v, \tag{2.3}$$

for an appropriate selection of a single τ for all poles. The matrix $(A-\tau I)^{-1}(A-\xi I)$ is known as a Cayley transformation in the eigenvalue context; see, e.g., [2]. Interestingly enough, this preconditioning approach has not appeared to have been investigated explicitly in this context, possibly because of the requirement to solve systems with the (real) matrix $A-\tau I$. We show in section 3 that this is precisely what the recently proposed method in [33], [41] performs when the exponential is replaced by a rational function. In section 4 we also show that the method in [1] amounts to solving (2.3) by first resorting to the normal equation and then applying a conjugate gradient (CG) method.

A seemingly different approach consists of approximating the operation $\exp(A)v$ by projecting the problem onto a subspace of a possibly much smaller dimension. Krylov subspaces have been extensively explored to this purpose, due to their favorable computational and approximation properties; see, e.g., [14], [15], [16], [20], [24], [25], [43]. Let $V_k = [v_1, \ldots, v_k]$ be an $n \times k$ matrix whose orthonormal columns span $K_k(A, v)$. The vectors v_i, $i = 1, \ldots, k$, can be generated by means of the following Lanczos recurrence:

$$AV_k = V_{k+1}H_{k+1,k} = V_k H_k + h_{k+1,k} v_{k+1} e_k^T, \qquad v = V_k e_1, \tag{2.4}$$

where e_i is the ith column of the identity matrix of a given dimension, e_k^T is the transpose of e_k, and $H_k = V_k^T A V_k$, $H_k = (h_{ij})$ is a symmetric tridiagonal matrix. An approximation to $x = \exp(A)v$ may be obtained as

$$x_k = V_k \exp(H_k) e_1. \tag{2.5}$$

We shall refer to this approximation as the "standard Lanczos" method. For $k \ll n$, the procedure projects the matrix problem onto the much smaller subspace $K_k(A, v)$, so that $\exp(H_k)$ can be cheaply evaluated with techniques such as scaling and squaring Padé [22]. The quality of the approximation strongly depends on the spectral properties of A and on the ability of $K_k(A, v)$ to capture them. Typically, convergence starts taking place after a number of iterations at least as large as $\|A\|^{1/2}$ [24]. A first characterization of the approximation was given by Saad in [34, Theorem 3.3], where it is shown that $V_k \exp(H_k)e_1$ represents a polynomial approximation $p(A)v$ to $\exp(A)v$, in which the polynomial p of degree $k-1$ interpolates the exponential in the Hermite sense on the set of eigenvalues of H_k. Other polynomial approximations have been explored; see, e.g., [13], [33].

It is important to realize that the partial fraction expansion and the Krylov subspace approximation may be related in a natural way whenever the exponential is replaced by a rational function. In such a situation, the two approaches may coincide

if, for instance, a Galerkin method is used to obtain the approximate solutions $x_k^{(j)}$. More generally, using (2.5) we can write

$$(2.6)\quad \left|\, \|\exp(A)v - x_k\| - \|\exp(A)v - V_k\mathcal{R}_\nu(H_k)e_1\|\,\right| \le \|\left(\exp(H_k) - \mathcal{R}_\nu(H_k)\right)e_1\|\,.$$

If $\mathcal{R}_\nu$ accurately approximates the exponential, then the two approaches that employ x_k and $V_k\mathcal{R}_\nu(H_k)e_1$ evolve similarly as the Krylov subspace dimension increases; see, e.g., the discussion in [19]. The behavior just described could justify the use of the partial fraction expansion in place of the standard method, especially if acceleration procedures can be determined to efficiently approximate each system solution. In fact, this is the unifying argument of the results in this paper.

3. The shift-invert Lanczos method. In [32] and independently in [41], the authors have proposed a technique for accelerating the standard Lanczos approximation to functions such as the exponential. The method is closely related to the shift-invert technique for eigenvalue problems and consists of first applying the Lanczos recurrence to the matrix $(I - \sigma A)^{-1}$, for some $\sigma \in \mathbb{R}$, $\sigma > 0$, and starting vector $v_1 = v$, $\|v\|$ is the 2-norm of v. giving

$$(3.1)\qquad (I - \sigma A)^{-1}V_m = V_m T_m + \beta_m v_{m+1} e_m^T.$$

An approximation to $y = \exp(A)v$ is then obtained as

$$(3.2)\qquad y_{SI} := V_m \exp((I - T_m^{-1})/\sigma)e_1.$$

The procedure in [32], [41] was tailored to general analytic functions f, and thus is perfectly applicable to the case of rational functions. For $f = \mathcal{R}_\nu$, in the next proposition we describe the shift-invert algorithmic procedure by means of a partial fraction expansion of $\mathcal{R}_\nu$. This allows us to analyze the selection of the shift parameter in [32], [41] for $f = \mathcal{R}_\nu$, that is, in terms of rational functions.

PROPOSITION 3.1. [illegible] $\mathcal{R}_\nu$ [illegible] $\mathcal{R}_\nu(z) = \omega_0 + \sum_{j=1}^{\nu} \omega_j/(z - \xi_j)$ [illegible] $\sigma > 0$ [illegible] y_{SI} [illegible] $y = \mathcal{R}_\nu(A)v$ [illegible] (3.2) [illegible] $\mathcal{R}_\nu$ [illegible] $y_{prec} = \omega_0 v + \sum_{j=1}^{\nu} \omega_j x_m^{(j)}$ [illegible] j $x_m^{(j)}$ [illegible] $x^{(j)} = (A - \xi_j I)^{-1}v$ [illegible] $K_m((A - \frac{1}{\sigma}I)^{-1}(A - \xi_j I), v)$ [illegible] $y_{SI} = y_{prec}$ [illegible]. When the exponential is replaced by $\mathcal{R}_\nu$ in (3.2), y_{SI} can be written as

$$(3.3)\qquad y_{SI} = V_m \left(\omega_0 e_1 + \sum_{j=1}^{\nu} \omega_j \left(-\frac{1}{\sigma} T_m^{-1} + \left(\frac{1}{\sigma} - \xi_j \right) I \right)^{-1} e_1 \right).$$

On the other hand, y_{prec} is obtained as $y_{prec} = \omega_0 v + \sum_{j=1}^{\nu} \omega_j x_m^{(j)}$, where each $x_m^{(j)}$ approximates $x^{(j)} = (A - \xi_j I)^{-1}v$. For $j = 1, \ldots, \nu$, we multiply by $(A - \frac{1}{\sigma}I)^{-1}$ the system $(A - \xi_j I)x^{(j)} = v$ from the left, that is

$$(3.4)\qquad \left(A - \frac{1}{\sigma} I \right)^{-1} (A - \xi_j I) x^{(j)} = \left(A - \frac{1}{\sigma} I \right)^{-1} v,$$

so that $(I + (\frac{1}{\sigma} - \xi_j)(A - \frac{1}{\sigma}I)^{-1})x^{(j)} = (A - \frac{1}{\sigma}I)^{-1}v$. We then determine $x_m^{(j)}$ by using a Galerkin procedure in $K_m((A - \frac{1}{\sigma}I)^{-1}(A - \xi_j I), v)$. Note that this space would not be the "natural" space for a standard Galerkin procedure, which would instead use

$K_m((A-\frac{1}{\sigma}I)^{-1}(A-\xi_j I),(A-\frac{1}{\sigma}I)^{-1}v)$, for left preconditioning. Due to the shift invariance of Krylov subspaces, it holds that

$$K_m\left(\left(A-\frac{1}{\sigma}I\right)^{-1}(A-\xi_j I),v\right)=K_m\left(\left(A-\frac{1}{\sigma}I\right)^{-1},v\right).$$

Moreover, relation (3.1) can be scaled as

$$\left(A-\frac{1}{\sigma}I\right)^{-1}V_m=-V_m\sigma T_m-\sigma\beta_m v_{m+1}e_m^T. \tag{3.5}$$

Therefore, let $x^{(j)}\approx x_m^{(j)}\in K_m((A-\frac{1}{\sigma}I)^{-1},v)$ with $x_m^{(j)}=V_m z_m^{(j)}$. Imposing the Galerkin condition on the residual vector yields

$$V_m^T\left(I+\left(\frac{1}{\sigma}-\xi_j\right)\left(A-\frac{1}{\sigma}I\right)^{-1}\right)V_m z_m^{(j)}=V_m^T\left(A-\frac{1}{\sigma}I\right)^{-1}V_m e_1.$$

Taking into account (3.5), we obtain

$$\left(I-\left(\frac{1}{\sigma}-\xi_j\right)\sigma T_m\right)z_m^{(j)}=-\sigma T_m e_1,$$

or, equivalently, $(-\frac{1}{\sigma}T_m^{-1}+(\frac{1}{\sigma}-\xi_j)I)z_m^{(j)}=e_1$. We have thus shown that

$$y_{prec}=V_m\left(\omega_0 e_1+\sum_{j=1}^{\nu}\omega_j\left(-\frac{1}{\sigma}T_m^{-1}+\left(\frac{1}{\sigma}-\xi_j\right)I\right)^{-1}e_1\right),$$

which is the same as (3.3). ∎

The previous proposition shows that when applied to a rational function, the shift-invert procedure is mathematically equivalent to a Galerkin procedure for the shifted systems involving the poles, appropriately preconditioned with the same matrix $(A-\frac{1}{\sigma}I)$. We will use this insightful relation to derive an automatic selection of the acceleration parameter σ.

3.1. Selecting the acceleration parameter. The effectiveness of the described scheme strongly depends on the choice of the acceleration parameter. In [41], an analysis is performed to select an optimal parameter at each iteration m, and the actual values are tabulated (cf. Table 3.1) by [illegible] evaluating the quantity

$$E_{m-1}^{m-1}(\sigma):=\inf_{r\in\Pi_{m-1}^{m-1}}\sup_{t\geq 0}|r(t)-\exp(-t)|,$$

where $\Pi_i^j=\{p(t)(1+\sigma t)^{-i}\,|\,p\in\Pi_j\}$ and Π_j is the space of polynomials of degree j or less. We stress that the inf-sup problem above depends on m, the iteration index. Our fully algebraic analysis aims to overcome this difficulty by resorting to rational functions in place of exp. Practical guidelines on how to use the tabulated values without varying the parameter at each iteration are also given in [41]. In [30], the author essentially conforms to this strategy. In both cases the employed arguments are tied to the theoretical analysis performed in [36].

TABLE 3.1
Some of the tabulated values in [41] *of the shift-invert parameter. m is the number of SI iterations.*

m	$E_m^m(\sigma_{opt})$	σ_{opt}	m	$E_m^m(\sigma_{opt})$	σ_{opt}
2	$2.0 \cdot 10^{-2}$	$4.93 \cdot 10^{-1}$	12	$1.6 \cdot 10^{-6}$	$1.19 \cdot 10^{-1}$
4	$3.1 \cdot 10^{-3}$	$1.75 \cdot 10^{-1}$	14	$2.5 \cdot 10^{-7}$	$8.64 \cdot 10^{-2}$
6	$4.0 \cdot 10^{-4}$	$1.91 \cdot 10^{-1}$	16	$4.0 \cdot 10^{-8}$	$8.67 \cdot 10^{-2}$
8	$6.5 \cdot 10^{-5}$	$1.90 \cdot 10^{-1}$	18	$6.6 \cdot 10^{-9}$	$6.78 \cdot 10^{-2}$
10	$9.7 \cdot 10^{-6}$	$1.19 \cdot 10^{-1}$	20	$1.1 \cdot 10^{-9}$	$6.82 \cdot 10^{-2}$

The result of Proposition 3.1 leads us to analyze the influence of the parameter σ with a completely different strategy, namely by studying its role in the preconditioned system (2.3), that is,

$$\left(I + \left(\frac{1}{\sigma} - \xi_j\right)\left(A - \frac{1}{\sigma}I\right)^{-1}\right) x^{(j)} = \left(A - \frac{1}{\sigma}I\right)^{-1} v. \tag{3.6}$$

In the rest of this section we omit the dependence of ξ_j and $x^{(j)}$ on j. Moreover, without loss of generality (cf. (2.1)), we consider only the complex poles with positive imaginary part.

We start by observing that the eigenvalues of the coefficient matrix are given by $\hat{\lambda} = 1 + (\frac{1}{\sigma} - \xi)/(\lambda - \frac{1}{\sigma})$, where λ is an eigenvalue of A; this means that the $\hat{\lambda}$'s lie on a line of the complex plane. Assuming that $\frac{1}{\sigma} - \xi \neq 0$ and dividing by $(\frac{1}{\sigma} - \xi)$, we obtain

$$\left(\left(A - \frac{1}{\sigma}I\right)^{-1} - \chi I\right) x = \tilde{v}, \quad \text{with} \quad \chi = \frac{1}{\xi - \frac{1}{\sigma}}, \tag{3.7}$$

and $\tilde{v}$ defined accordingly. The eigenvalues of the coefficient matrix lie on the horizontal line $(\hat{x}, y_0)$ with

$$y_0 := \frac{\Im(\xi)}{|\frac{1}{\sigma} - \xi|^2}, \quad \text{and} \quad \hat{x} \in \left[-\frac{1}{\frac{1}{\sigma}} + \frac{\frac{1}{\sigma} - \Re(\xi)}{|\frac{1}{\sigma} - \xi|^2}, \frac{1}{\alpha - \frac{1}{\sigma}} + \frac{\frac{1}{\sigma} - \Re(\xi)}{|\frac{1}{\sigma} - \xi|^2}\right].$$

The assumption $\frac{1}{\sigma} - \xi \neq 0$ is not restrictive: If $\frac{1}{\sigma} = \xi$ for some (real) ξ, then from (3.6) it follows that the system solution associated with that pole is readily obtained, and the analysis need not be performed.

The coefficient matrix in (3.7) is given by a real negative definite symmetric matrix shifted by a complex multiple of the identity. It was shown in [17], [27] that in this case the performance of Krylov subspace methods may be fully characterized by using spectral information of the coefficient matrix. Therefore, estimates for the optimal parameter σ may be obtained by analyzing the spectrum of $((A - \frac{1}{\sigma}I)^{-1} - \chi I)$ as σ changes. To this end, we recall here the following bound for the linear system error in our notation.

PROPOSITION 3.2 (see [27, Lemma 5.2]). [illegible] $(\widetilde{A} - \chi I)x = \tilde{v}$ [illegible] $\widetilde{A}$ [illegible] $\chi \in \mathbb{C}$, [illegible] x_m [illegible] $K_m(\widetilde{A}, \tilde{v})$ [illegible] $\lambda_{\max}, \lambda_{\min}$ [illegible] $\widetilde{A} - \Re(\chi)I$ [illegible]

$$||x - x_m|| < g(\lambda_{\min}, \lambda_{\max}, \tilde{v}, \chi)\ \frac{1}{\rho^m + 1/\rho^m},$$

... g ... $\widetilde{A}, \tilde{v}$ ... χ ... $\rho = \gamma + \sqrt{\gamma^2 - 1}$ ...

$$\gamma = \frac{|\lambda_{\min} - i\Im(\chi)| + |\lambda_{\max} - i\Im(\chi)|}{|\lambda_{\min} - \lambda_{\max}|}.$$

The proposition above shows that the larger γ, the faster the convergence in terms of the subspace dimension m. We recall that $\mathrm{spec}(A) \subset [\alpha, 0]$ with $\alpha < 0$. In our context, we can apply the result above both to the original partial fraction expansion approximation, as well as to the preconditioned system (3.6). In the former case, setting $\widetilde{A} = A$ and $\chi = \xi$, we obtain

$$\gamma(\xi) = \frac{|\alpha - \xi| + |\xi|}{-\alpha}. \tag{3.8}$$

In the preconditioned case, setting $\widetilde{A} = (A - \frac{1}{\sigma} I)^{-1}$ and $\chi = 1/(\xi - \frac{1}{\sigma})$, after simple algebraic manipulations, we get

$$\gamma^{prec}(\xi, \sigma) = \frac{(\frac{1}{\sigma} - \alpha)|\xi| + \frac{1}{\sigma}|\alpha - \xi|}{-\alpha|\frac{1}{\sigma} - \xi|}. \tag{3.9}$$

The expression in (3.8) shows that for $|\alpha| \gg |\xi|$, the error bound predicts very slow convergence of the linear system, as in this case $\gamma \approx 1$. It is desirable that a well-chosen σ make $\gamma^{prec}(\xi, \sigma)$ much larger than $\gamma(\xi)$, so as to significantly improve the convergence rate. An ideal value of σ would satisfy something like

$$\min_{\xi} \gamma^{prec}(\xi, \sigma) \geq \max_{\xi} \gamma(\xi),$$

to ensure faster convergence for all poles. However, this inequality turned out to be hard to analyze. Nonetheless, it is possible to relate the two convergence coefficients. To simplify the notation, in the rest of this section we use

$$\tau := \frac{1}{\sigma},$$

and, with some abuse of notation, we use $\gamma^{prec}(\xi, \tau)$. We have

$$\gamma^{prec}(\xi, \tau) = F(\alpha, \xi, \tau)\gamma(\xi), \tag{3.10}$$

where

$$F(\alpha, \xi, \tau) = \frac{\tau}{|\tau - \xi|} - \frac{\alpha|\xi|}{(|\alpha - \xi| + |\xi|)|\tau - \xi|} = \frac{\tau - c}{|\tau - \xi|}, \tag{3.11}$$

with $c = \alpha|\xi|/(|\alpha - \xi| + |\xi|)$.

The following proposition shows that, for each pole, it is possible to determine the least value of the parameter that improves convergence, and also the one that maximizes the ratio between the two convergence rates. Unfortunately, the resulting parameter depends on the given pole, and thus it may not be optimal for other poles.

PROPOSITION 3.3. ... α ... ξ ... $F(\tau) = F(\alpha, \xi, \tau)$ ... (3.11) ... ... $\Re(\xi) > \alpha/2$ ... $\Im(\xi) \neq 0$ ...

(i) $F(\tau) \geq 1$ ... $\tau \geq \tau_0$ ... $\tau_0 = \frac{1}{2} \frac{|\xi|^2 - c^2}{\Re(\xi) - c}$ ... $\Re(\xi) > c$

(ii) $F(\tau_{\max}) \geq F(\tau)$ [illegible] τ [illegible]

$$\tau_{\max} = \tau_{\max}(\xi) = \frac{\Re(\xi)c - |\xi|^2}{c - \Re(\xi)} \qquad F(\tau_{\max}) = \frac{|c-\xi|}{|\Im(\xi)|} \geq 1; \tag{3.12}$$

(iii) $\gamma^{prec}(\xi, \tau_0) = \gamma(\xi)$ [illegible] $\lim_{\tau\to\infty} \gamma^{prec}(\xi, \tau) = \gamma(\xi)$

[illegible]. Let $\xi = \xi_R + \imath\xi_I$. We first show that $\xi_R > c$. Since $c < 0$, then clearly $\xi_R > c$ when $\xi_R \geq 0$. For $\xi_R < 0$, using $\alpha < 2\xi_R$ we obtain $\alpha|\xi| < 2\xi_R|\xi| \leq \xi_R|\xi| \leq \xi_R(|\xi| + |\alpha - \xi|)$, from which

$$\xi_R > \frac{\alpha|\xi|}{|\xi| + |\alpha - \xi|} = c.$$

To prove (i), we observe that

$$F(\tau) \geq 1 \qquad \Leftrightarrow \qquad 2(\xi_R - c)\tau \geq |\xi|^2 - c^2. \tag{3.13}$$

Using $\xi_R > c$, the previous requirement corresponds to imposing $\tau \geq \tau_0$.

To prove (ii) we explicitly write

$$F'(\tau) = -\frac{(\tau - \xi_R)}{|\tau - \xi|^3}(\tau - c) + \frac{1}{|\tau - \xi|} = 0 \quad \Leftrightarrow \quad -(\tau - \xi_R)(\tau - c) + |\tau - \xi|^2 = 0,$$

from which the expression for $\tau_{\max}$ follows. Moreover, F is an increasing function for $\tau \leq \tau_{\max}$ and a decreasing one otherwise, so that $F(\tau_{\max})$ is a maximum.

To prove that $F(\tau_{\max}) \geq 1$ we notice that $F(\tau_{\max})^2 = 1 + (c - \xi_R)^2/\xi_I^2$, from which we obtain that $(F(\tau_{\max}) - 1)(F(\tau_{\max}) + 1) = \frac{(c-\xi_R)^2}{\xi_I^2}$. The result follows by taking into account that

$$F(\tau_{\max}) + 1 = \frac{|\xi_I| + |c - \xi|}{|\xi_I|}.$$

Finally, the first equality in (iii) follows from $F(\tau_0) = 1$ in (3.13), while it can be readily verified that $\lim_{\tau\to\infty} F(\tau) = 1$. ∎

In light of Proposition 3.3(i), one could restrict the choice of the parameter τ to the interval $[\tau_0, \infty[$. However, (iii) indicates that values of the parameter that are too close to the extremes of this interval do not accelerate convergence; see similar conclusions in [30]. The hypothesis that $\Re(\xi) > \alpha/2$ is crucial; otherwise $F(\tau) \geq 1$ only for $\tau < 0$. The only (unlikely) exception is $\xi = \alpha/2 \in \mathbb{R}$, in which case $F(\tau) \geq 1$ for any nonnegative τ. On the other hand, for the values of α of interest, $|\alpha| \gg |\xi|$, and thus the hypothesis $\Re(\xi) > \alpha/2$ is clearly verified. It is also important to notice that for $|\alpha| \gg |\xi|$ it follows that $\tau_{\max} \approx |\xi|$, indicating the obvious fact that, for each pole ξ, the best real parameter is related to the pole itself.

Although quite sharp, the results above still depend on the spectrum of A through α and do not provide a simple way to select a good single parameter for all poles. To complete our understanding, we thus look for a quantity that well represents the behavior of γ^{prec}, especially for large $|\alpha|$. To this end, we observe that $-c \leq |\xi|$, so that $F(\xi, \tau) \leq \frac{\tau+|\xi|}{|\tau-\xi|}$. The bound is sharp for $|\alpha| \gg |\xi|$, that is, $\gamma(\xi) \approx 1$, in which case $c \approx -|\xi|$. The quantity $\mathcal{H}(\tau, \xi) := \frac{\tau+|\xi|}{|\tau-\xi|} \geq 1$ also appears explicitly in the following lower bound for γ^{prec}:

$$\gamma^{prec}(\tau, \xi) = \frac{\tau}{|\tau - \xi|}\gamma(\xi) + \frac{|\xi|}{|\tau - \xi|} \geq \frac{\tau}{|\tau - \xi|} + \frac{|\xi|}{|\tau - \xi|} = \mathcal{H}(\tau, \xi), \tag{3.14}$$

TABLE 3.2
Values of $\mathcal{H}(\tau_i, \xi_j)$, $i, j = 1, \ldots, \nu$, for Chebyshev with $\nu = 14$ (complex conjugates are not shown).

$\tau_i = \lvert\xi_i\rvert$	ξ_1	ξ_3	ξ_5	ξ_7	ξ_9	ξ_{11}	ξ_{13}
18.8616	1.1657	1.2516	1.3564	1.4831	1.6260	1.7628	1.8515
14.1496	1.1615	1.2590	1.3905	1.5708	1.8105	2.0910	2.3111
10.9932	1.1515	1.2533	1.4010	1.6254	1.9739	2.4925	3.0433
8.7609	1.1387	1.2391	1.3924	1.6430	2.0832	2.9193	4.3218
7.2115	1.1261	1.2219	1.3727	1.6300	2.1170	3.2233	6.5221
6.2274	1.1160	1.2068	1.3520	1.6045	2.0975	3.3081	8.9488
5.7485	1.1105	1.1981	1.3391	1.5859	2.0716	3.2821	9.5758

and this estimate is again sharp when $\gamma(\xi) \approx 1$. We next analyze the behavior of $\mathcal{H}$, which does not depend on the spectrum of A, but only on the poles and on the parameter. We have

$$\mathcal{H}(\tau, \xi)^2 = 1 + 2(|\xi| + \Re(\xi))\frac{\tau}{|\tau - \xi|^2}.$$

Note that $|\xi| + \Re(\xi) \geq 0$ for any ξ, and for a given nonreal pole ξ it holds that $\frac{\tau}{|\tau-\xi|^2} \leq \frac{|\xi|}{||\xi|-\xi|^2}$, where the right-hand side is attained for $\tau = |\xi|$. Therefore, for each pole ξ_i,

$$\mathcal{H}(|\xi_i|, \xi_i) = \max_{\tau > 0} \mathcal{H}(\tau, \xi_i).$$

To take τ_i as a priori parameter for all systems, we need to make sure that this value of τ is also effective for a different pole ξ_j. Let the poles be sorted with decreasing (positive) imaginary parts. Setting $\tau_i = |\xi_i|$, we state the following discrete problem:[2]

$$\max_{\tau_1, \ldots, \tau_\nu} \min_{\xi_1, \ldots, \xi_\nu} \mathcal{H}(\tau_i, \xi_j), \tag{3.15}$$

which can be solved, once and for all, for a given class of rational functions and for each selected degree. As an example, Table 3.2 reports the values of $\mathcal{H}(\tau_i, \xi_j)$ as τ_i and ξ_j vary, for Chebyshev rational functions and $\nu = 14$ (poles are computed from the coefficients as listed in [9]). In the table, the optimal value of τ for problem (3.15) with $\nu = 14$ is given by $\tau_1 = 18.8616$, ensuring that $\gamma^{prec}(\xi, \tau) \geq \mathcal{H}(\tau_1, \xi_1) = 1.1657$. Note that, for all degrees, the best value of τ turns out to always be associated with ξ_1. Therefore, we propose to use the parameter

$$\tau_{\text{opt}} := |\xi_1| \quad \Leftrightarrow \quad \sigma_{\text{opt}} = \frac{1}{|\xi_1|}. \tag{3.16}$$

The corresponding values associated with Chebyshev rational poles are listed in Table 3.3 for $\nu \leq 20$. The entries in the table can be used as follows: If a final tolerance *tol* on the approximation of $\exp(A)v$ is requested, then the shift-invert approach may be used with a shift value corresponding to $\nu \geq -\log_{10}(tol)$ (e.g., $tol = 10^{-8}$ yields $\nu \geq 8$ so that $\sigma = 0.1062$ or a smaller value in the table may be used).

Our derivation suggests a parameter selection somehow similar to that given in [41] (cf. Table 3.1), although our justification is completely different, and it does not depend on m. This similarity may be viewed as an additional motivation for the reliability of the approach.

[2] A continuous problem in τ and ξ could also be formulated, but the unnecessary added difficulty is beyond the scope of this analysis.

TABLE 3.3
Optimal values of the parameter (cf. (3.16)*) for various rational function degrees.*

ν	1	2	3	4	5	6	7	8	9	10
$\sigma_{\rm opt}$	1.7271	0.7565	0.4134	0.2720	0.1988	0.1551	0.1264	0.1062	0.0914	0.0801
ν	11	12	13	14	15	16	17	18	19	20
$\sigma_{\rm opt}$	0.0711	0.0639	0.0580	0.0530	0.0488	0.0452	0.0421	0.0394	0.0369	0.0348

3.2. Asymptotic behavior. Our parameter selection is based on asymptotic arguments, that is, on information of the matrix spectral interval and not on the actual eigenvalue distribution. In particular, we recall that the convergence of (symmetric) linear systems often exhibits superlinear behavior, in the sense that the rate of convergence may increase as convergence takes place; see, e.g., the discussion in [39]. Such important characterization is not captured by an asymptotic analysis. Therefore, in some cases other values of the parameter may lead to better convergence than that obtained with our analytically selected choice. As an example, we consider the matrix $\widetilde{A}$ of size $n = 3375$ of Example 5.1 in section 5, whose extreme eigenvalues are $\lambda_{\min} \approx -2329.4$ and $\lambda_{\max} \approx -22.597$, and we define the singular matrix $A = \widetilde{A} - \lambda_{\max} I$. We study the performance of the accelerated process with the optimal parameter $\sigma_{\rm opt} = 0.053$ and with another possible candidate, $\sigma_{\min} = 1/\max_j |\Re(\xi_j)| = 0.1124$, taken for $\nu = 14$ poles. The vector v is taken as a normalized vector of all ones. Figure 3.1 shows the convergence curves of the SI procedure with A and the two parameters (lower solid and dashed curves), showing a slightly better performance of $\sigma_{\min}$ over $\sigma_{\rm opt}$; this is not predicted by our theory. However, our arguments better describe the behavior of the $n \times n$ diagonal matrix D (middle solid and dashed curves), whose nonzero entries are [illegible] values in the same spectral interval as A. The vector v is unchanged. In this case, the convergence slope is steeper when using $\sigma_{\rm opt}$ than with $\sigma_{\min}$. The upper curves show the convergence rate predicted by the asymptotic quantity $\mathcal{H}(1/\sigma, \xi_1)^j$, $j = 1, \ldots, m$, for $\sigma = \sigma_{\rm opt}$ (filled squares) and $\sigma = \sigma_{\min}$ (circles). Note that both curves well represent the initial convergence phase of the shift-invert

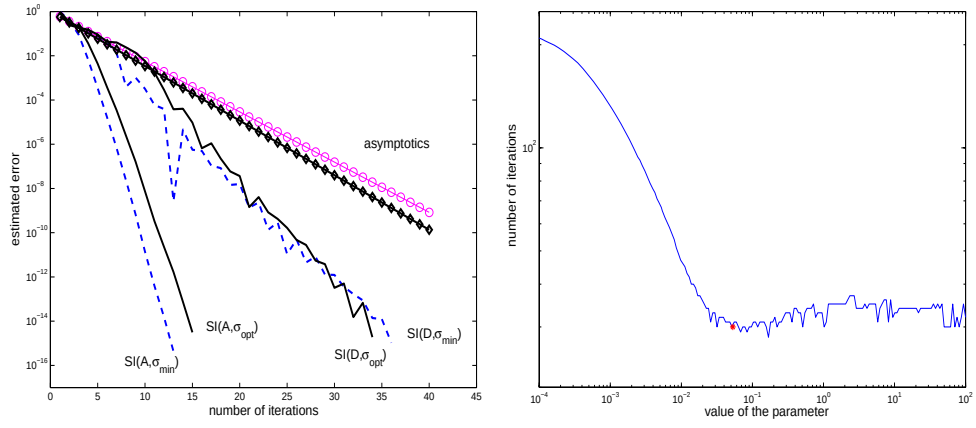


FIG. 3.1. *Left: Convergence history of SI for a matrix stemming from a shifted Laplace operator and for a diagonal matrix D with uniformly distributed eigenvalues in* $\mathrm{spec}(A)$. *Here* $\sigma_{\rm opt} = 0.053$ *and* $\sigma_{\min} = 0.1124$. *Also reported are the asymptotic values* $\mathcal{H}(1/\sigma, \xi_1)^j$, $j = 1, \ldots, m$, *for* $\sigma_{\min}$ *(circles) and* $\sigma_{\rm opt}$ *(filled squares). Right: Number of iterations of SI applied to the diagonal matrix D versus value of the shift σ; the symbol "*" refers to the choice* $\sigma = \sigma_{\rm opt}$.

procedure with D, with a slightly better performance for $\mathcal{H}(1/\sigma_{\text{opt}}, \xi_1)$. To fully appreciate the performance of the choice $\sigma = \sigma_{\text{opt}}$ with the matrix D, in the right plot of Figure 3.1 we display the number of iterations of SI to achieve the required stopping threshold, as the value of the parameter varies in $[10^{-4}, 10^2]$; the symbol "*" refers to the choice $\sigma = \sigma_{\text{opt}}$. Note that the performance of the method is not overly sensitive to overshooting values of σ (and this conforms to the tabulated values in Table 3.1), but it may considerably degrade if σ is chosen too small. In particular, the plot shows that a typical practical value suggested in [41, section 6], namely $\sigma = 0.01$, would force the method to perform more iterations on this matrix.

3.3. Implementation details. The algorithmic aspects of the shift-invert procedure were described in [41]. A possible implementation generates the matrix V_m one vector at a time by means of the Lanczos algorithm (see, e.g., [21]) and the corresponding elements of the tridiagonal matrix T_m in (3.1). It is important to observe that convergence at high accuracy is often obtained for a small approximation space, so that little memory is required to store V_m. The difficulties associated with the approximate solutions with $I - \tau A$ are also treated in [41].

A crucial part in the overall procedure is how to monitor convergence, since the error norm is not available. Although the analysis of stopping criteria is beyond the scope of this paper, we need to face this problem to avoid premature termination; we refer to [19] for a recent analysis and an accurate estimation of the error norm for the approximation of various rational matrix operators. With the notation of (3.5), a classical stopping criterion is given by the quantity

$$t_{m+1,m} |e_m^T \exp((I - T_m^{-1})/\sigma) e_1|, \tag{3.17}$$

which is cheaply available during the computation; in the case of standard Lanczos, using (2.5) the criterion above reduces to $h_{k+1,k} |e_k^T \exp(H_k) e_1|$. It is known that for m very small, this quantity may highly underestimate the true error; cf. the (red) dash-dotted curve of Figure 3.2. In our experiments of sections 5 and 6, for the first few iterations we replace the absolute estimate of the error with a relative quantity, until this falls below a safeguard parameter set to 10^{-2}. In the case of (3.17), this

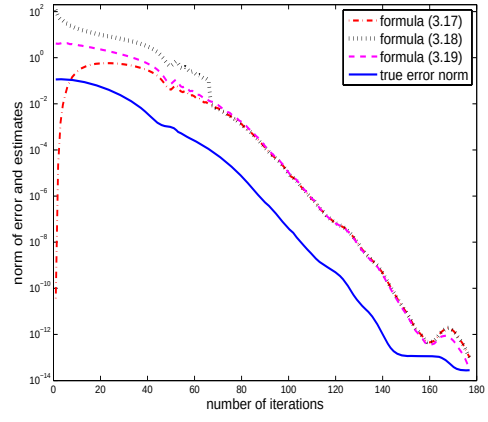


FIG. 3.2. *Convergence history of standard Lanczos and different error estimates. The safeguard parameter for* (3.18) *is equal to* 0.1 *(see text).*

quantity reads as

$$t_{m+1,m}|e_m^T \exp((I - T_m^{-1})/\sigma)e_1|/|e_1^T \exp((I - T_m^{-1})/\sigma)e_1|. \tag{3.18}$$

In practice, this somewhat conservative safeguard procedure is maintained until the components of the approximation vector $\exp((I - T_m^{-1})/\sigma)e_1$ take the expected exponential pattern; see [26]. The reported values in the figure refer to the approximation of $\exp(0.1A)v$, where A is the 4900×4900 matrix stemming from the two-dimensional (2D) Laplace operator with Dirichlet homogeneous boundary conditions, v is the normalized vector of all ones, and the safeguard parameter is equal to 0.1. In Figure 3.2 we also report the behavior of the higher-order estimate (for standard Lanczos)

$$t_{k+1,k}|e_k^T \varphi_1(H_k)e_1|, \quad \varphi_1(z) = \frac{\exp(z) - 1}{z}, \tag{3.19}$$

which was also proposed in [34]. We observe that this estimate seems to cure the problem given by the lower-order estimate in (3.17). In practice, whenever $e_k^T \exp(H_k)e_1$ is too small, so that (3.17) is unreliable, it holds that $|e_k^T \varphi_1(H_k)e_1| \approx |e_k^T H_k^{-1} e_1|$. From Figure 3.2 it is clear that both the safeguard strategy and the higher-order criterion allow one to safely continue the iteration until true convergence takes place. In our experiments we used (3.18) because it is in general cheaper to compute than (3.19). We refer to [12], [19], [27], [34] for further considerations and for higher-order stopping criteria.

4. Application of a real-valued method for solving linear systems. In this section we increase our understanding of a method recently proposed by Axelsson and Kucherov in [1], for solving complex symmetric systems by means of a formulation that only uses real arithmetic computation. The method can be used in our context when a partial fraction expansion of a rational function approximation is employed, as described in section 2. We show that the method can be derived using our preconditioning framework. Moreover, we propose a variant that makes the overall procedure significantly more efficient.

We first briefly recall the main steps of the approach. Given the complex system

$$(R + \imath S)u = b, \tag{4.1}$$

with $u = u_R + \imath\, u_I$ and $b = b_R + \imath\, b_I$, the proposed procedure uses the real form

$$\begin{pmatrix} R - \eta S & \sqrt{1+\eta^2}S \\ \sqrt{1+\eta^2}S & -R - \eta S \end{pmatrix} \begin{pmatrix} u_R \\ z \end{pmatrix} = \begin{pmatrix} b_R \\ (b_I - \eta b_R)/(\sqrt{1+\eta^2}) \end{pmatrix},$$

where $\eta > 0$ is a real parameter and $z = (\eta u_R - u_I)/\sqrt{1+\eta^2}$. The Schur complement reduction provides the following linear system for u_R

$$Cu_R = w \tag{4.2}$$

with $C = R - \eta S + (1+\eta^2)S(R + \eta S)^{-1}S$ and $w = b_R + S(R + \eta S)^{-1}(b_I - \eta b_R)$. The imaginary part, u_I, may be computed by using the relation $R\, u_R - S\, u_I = b_R$. It is shown in [1] that under certain hypotheses on S and R, it is possible to derive an optimal choice of η so that the matrix $B = R + \eta S$ is an effective preconditioner for the system (4.2).

In our context, the complex symmetric system to be solved is $(A-\xi I)u = v$ for a fixed pole $\xi = \xi_R + \imath\xi_I$. Therefore, we have $R = A - \xi_R I$ and $S = -\xi_I I$. Substituting in the coefficient matrix of the system (4.2) we obtain

$$C = -B + 2\eta\,\xi_I\, I - (1+\eta^2)\xi_I^2 B^{-1},$$

where the preconditioner becomes $B = -(R+\eta S) = (\xi_R + \eta\,\xi_I)I - A$, and the preconditioned system reads

$$Mu_R = B^{-1}w, \quad \text{with} \quad \begin{aligned} M &= B^{-1}C = -I + 2\eta\xi_I B^{-1} - \xi_I^2(1+\eta^2)B^{-2}, \\ w &= v - \eta\,\xi_I B^{-1} v. \end{aligned} \tag{4.3}$$

Moreover, $u_I = \frac{1}{\xi_I}(-A+\xi_R I)u_R + \frac{1}{\xi_I}v$. Therefore, for each pole ξ, the original complex symmetric system is transformed into the real (preconditioned) system (4.3), which needs to be solved by an iterative method. We next show in Proposition 4.1 that the matrix $-M$ is symmetric positive definite for any choice of $\eta > 0$ and for all poles, and thus the conjugate gradient method can be used. Moreover, we show that the system (4.3) resulting from the procedure outlined above is nothing but the real part of the normal equation of (2.3) for a special choice of the acceleration parameter.

PROPOSITION 4.1. [illegible] u_R [illegible] $Mu_R = B^{-1}w$ (4.3) [illegible] $\tau = \xi_R + \eta\,\xi_I$, $\eta > 0$, [illegible] the real part of the normal equation

$$(\tau I - A)^{-1}(A - \xi I)u = (\tau I - A)^{-1}v,$$

[illegible] $K = (\tau I - A)^{-1}(A - \xi I)$ [illegible] $M = -K^*K \in \mathbb{R}^{n\times n}$ [illegible] u_R [illegible] $K^*Ku = K^*(\tau I - A)^{-1}v$ [illegible]. Let $R = A - \xi_R I$ and $S = -\xi_I I$, and note that R and S commute so that

$$\begin{aligned} K^*K &= (R+\eta S)^{-2}(R - iS)(R + iS) = (R+\eta S)^{-2}(R^2 + S^2) \\ &= I + 2\eta\xi_I(R+\eta S)^{-1} + \xi_I^2(1+\eta^2)(R+\eta S)^{-2} = -M. \end{aligned}$$

Therefore, K^*K is real symmetric and M is negative definite. Analogously, we can write $K^*(\tau I - A)^{-1}v = (R+\eta S)^{-2}(R - iS)v$ whose real part is given by $\Re(K^*(\tau I - A)^{-1}v) = (R+\eta S)^{-2}Rv = (R+\eta S)^{-1}w$. Therefore, the real part of the equation $K^*Ku = K^*(\tau I - A)^{-1}v$ is given by $-M\Re(u) = -B^{-1}w$, from which it follows that $u_R = \Re(u)$. ∎

The original implementation in [1] provided an optimal parameter τ for each shifted system to be solved, yielding a different acceleration matrix $(\tau I - A)^{-1}$ for each pole. The authors suggested using $\tau = \tau(\xi_j) = |\xi_j|$, for $A - \xi_j I$ having definite symmetric real part. This condition is not satisfied in our case, since $\Re(\xi_j)$ may be either positive or negative. In the next section we show how to select a single τ for all systems, so as to lower the computational costs.

A completely different preconditioning strategy could also be adopted. The use of an optimal preconditioner $(\tau I - A)$ with $\tau = \tau(\xi)$ would be feasible if it were possible to update the factorization for different shifts without recomputing the factors from scratch; see the results in [6] in this direction for linear system preconditioning.

4.1. Selecting the acceleration parameter. In this section we derive a single quasi-optimal positive parameter τ, from which, according to the relation $\tau = \xi_R + \eta\xi_I$, a different η follows for each system in (4.3). Therefore, while M differs for each shifted system, the matrix $B = \tau I - A$ is the same for all shifts.

Proposition 4.1 shows that M is symmetric and negative definite for any $\eta > 0$. Similar conclusions were derived in [1, Remark 1]. The next proposition provides sharp bounds for the condition number of M with no further hypotheses on $A - \xi I$.

PROPOSITION 4.2. [illegible] 4.1 [illegible] $\tau > \max\{0, \Re(\xi)\}$ [illegible]

$$\text{cond}(M) \le \max\left\{\frac{|\xi|^2}{\tau^2}, \frac{|\alpha-\xi|^2}{(\alpha-\tau)^2}\right\}\frac{|\tau-\xi|^2}{\xi_I^2}. \tag{4.4}$$

[illegible] $\tau \le |\xi|$ [illegible] $\frac{|\alpha-\xi|^2}{(\alpha-\tau)^2} \le \frac{|\xi|^2}{\tau^2}$ [illegible]

$$\text{cond}(M) \le \frac{|\xi|^2}{\xi_I^2}\frac{|\tau-\xi|^2}{\tau^2}. \tag{4.5}$$

[illegible]. Writing $-M = (R+\eta S)^{-2}(R^2+S^2) = (R-\eta\xi_I I)^{-2}(R^2+\xi_I^2 I)$, we get

$$\text{spec}(-M) = \left\{\frac{(\lambda-\xi_R)^2+\xi_I^2}{(\lambda-\tau)^2}\middle|\lambda \in \text{spec}(A)\right\}.$$

For $\lambda \in [\alpha, 0]$, let $\mu \in \text{spec}(-M)$, $\mu = g(\lambda) = \frac{\lambda^2-2\lambda\xi_R+|\xi|^2}{(\lambda-\tau)^2}$. We have

$$g'(\lambda) = 2\frac{\lambda(\xi_R-\tau)+\tau\xi_R-|\xi|^2}{(\lambda-\tau)^3} = 0 \quad \Leftrightarrow \quad \hat{\lambda} := \frac{\tau\xi_R-|\xi|^2}{\tau-\xi_R}.$$

Since $\tau > \xi_R$, it holds that $g'(\lambda) > 0$ only for $\lambda > \hat{\lambda}$; hence

$$g(\hat{\lambda}) = \frac{\xi_I^2}{|\tau-\xi|^2} \le \mu \quad \forall \mu \in \text{spec}(-M). \tag{4.6}$$

To derive an upper bound, we notice that, since $\hat{\lambda}$ is the only critical point and it is associated with a minimum, the maximum of g in $[\alpha, 0]$ is given by $\max\{g(\alpha), g(0)\}$. Collecting this bound and (4.6), the bound (4.4) for $\text{cond}(M)$ follows.

We next assume that $\tau \le |\xi|$ holds for all poles ξ. We write

$$g(\alpha) - g(0) = \frac{\alpha^2(\tau^2-|\xi|^2) - 2\alpha\tau(\xi_R\tau-|\xi|^2)}{\tau^2(\tau-\alpha)^2}.$$

For $\tau \le |\xi|$ the first addend in the numerator of the last expression is negative. For the second addend, we separately treat the cases of positive and negative pole's real part. If $\xi_R < 0$, then the second addend gives $-2\alpha\tau(\xi_R\tau - |\xi|^2) \le 0$. If $\xi_R > 0$, then we can get $-2\alpha\tau(\xi_R\tau-|\xi|^2) \le -2\alpha\tau(\tau^2-|\xi|^2) \le 0$. We have thus shown that $g(\alpha) - g(0) \le 0$, which completes the proof. ∎

The bound in (4.4) may be rather sharp. Its sharpness depends on whether the extremes of the function g defined in the proof are attained. Table 4.1 reports the bound in (4.4) for the 125×125 matrix obtained by the discretization of the three-dimensional (3D) Laplacian with homogeneous boundary conditions, shifted so as to have zero largest eigenvalue. The poles correspond to the Chebyshev rational approximation of degree $\nu = 14$. We used $\tau_{\text{opt}} = \min_{j=1,\ldots,\nu}|\xi_j| = 5.7485$; see below for an explanation of this choice.

TABLE 4.1
Condition number of $M = M(\xi_j)$ and its upper bound in (4.4), as the poles vary.

ξ_j ($\nu = 14$)	cond(M)	Estim.	ξ_j ($\nu = 14$)	cond(M)	Estim.
-8.8978 + 16.631i	19.115	19.115	-3.7033 + 13.656i	8.960	8.960
-0.2087 + 10.991i	4.717	4.731	2.2698 + 8.4617i	2.701	2.715
3.9934 + 6.0048i	1.700	1.708	5.0893 + 3.5888i	1.212	1.213
5.6231 + 1.1941i	1.009	1.011			

To derive a single parameter τ for all poles ξ, we study the bound in (4.5), which does not depend on the spectrum of A. We will see that for the Chebyshev approximation it is possible to derive a single parameter that satisfies $\tau \leq |\xi|$.

Let $W_\xi(\tau) = \frac{|\tau-\xi|^2}{\tau^2}$ be the part of the upper bound in (4.5) that depends on τ. It can be verified that $W_\xi(\tau)' = -\frac{2}{\tau^3}(-\tau\Re(\xi) + |\xi|^2)$, so that

$$W_\xi(\tau)' = 0 \quad \Leftrightarrow \quad \tau_*(\xi) = \Re(\xi) + \frac{\Im(\xi)^2}{\Re(\xi)} = \frac{|\xi|^2}{\Re(\xi)}.$$

If $\Re(\xi) < 0$, then $W_\xi(\tau_*)$ is a maximum and $\tau_*(\xi)$ is negative. We thus restrict our attention to the poles with positive real parts.[3] Moreover, we observe that, for $\tau > \tau_*$ and $\Re(\xi) < 0$, the function W_ξ is decreasing, so that the larger τ, the smaller the bound for $\Re(\xi) < 0$. We then recall that for (4.5) to hold, the selected parameter τ must satisfy

$$\Re(\xi) \leq \tau \leq |\xi| \quad \forall \xi.$$

Let the poles be sorted as $\Re(\xi_1) \leq \cdots \leq \Re(\xi_\nu)$. Then $\tau_*(\xi_\nu) \geq \Re(\xi_j)$ for $j \leq \nu$, and we define

$$\tau_{\text{opt}} := \min\left\{ \min_{j=1,\ldots,\nu} |\xi_j|, \tau_*(\xi_\nu) \right\}. \tag{4.7}$$

For the Chebyshev poles it holds that $\min_{j=1,\ldots,\nu} |\xi_j| = |\xi_\nu|$ so that

$$\tau_{\text{opt}} = |\xi_\nu|.$$

In Figure 4.1 we report the total number of conjugate gradient iterations required by the method to solve all systems $Mu_R = \hat{w}$ (see Algorithm AK), for different values of the parameter $\tau \in [0, 7]$. The data are as in Example 5.1 and $\nu = 8$. The symbol "*" indicates the number of iterations for the choice $\tau = \tau_{\text{opt}}$, showing the high quality of the a priori selected parameter.

The analysis above conforms with the multiple choice in [1], although in our case extremely fast convergence cannot be achieved for ,, shifted systems. It is also interesting that, as opposed to the shift-invert procedure, the pole with the , ,, , modulo is selected as the optimal parameter.

4.2. Implementation details. The real-valued method for approximating $y = \exp(A)v$ can be summarized as follows. For simplicity and without loss of generality, we take here a rational function of even degree. For odd degree rational approximation, the real shifted system corresponding to the real pole can be solved explicitly without resorting to the method discussed previously.

[3] We recall that in the case of Chebyshev approximation, poles are used with the opposite sign.

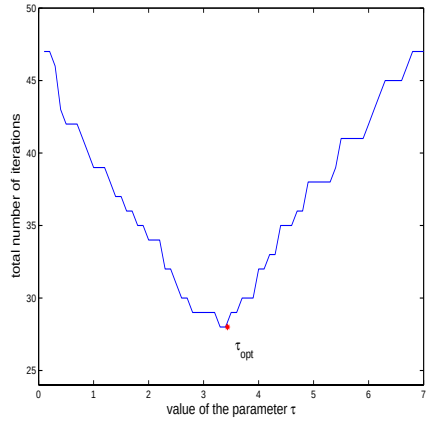


FIG. 4.1. *Total number of iterations for the variant of the Axelsson–Kucherov method, as a function of the parameter τ. The symbol "*" refers to the number of iterations for $\tau = \tau_{\text{opt}}$.*

ALGORITHM AK.

Given A, v, $\xi_1, \ldots, \xi_\nu$, $\omega_1, \ldots, \omega_\nu$

(i) Choose a parameter $\tau > 0$.

(ii) Set $B = \tau I - A$; $w_1 = B^{-1}v$; $w_2 = B^{-1}w_1$.

(iii) For each pole $\xi_j = \xi_R + \imath\xi_I$, $j = 1, 3, 5, \ldots, \nu - 1$:

- Solve $Mu_R = \widehat{w}$ with $M = -I + 2(\tau - \xi_R)B^{-1} - |\tau - \xi|^2 B^{-2}$ and $\widehat{w} = +w_1 - (\tau - \xi_R)w_2$.
- Compute $u_I = \frac{1}{\xi_I}(-Au_R + \xi_R u_R + v)$.
- Set $x_j = u_R + \imath u_I$.

(iv) Compute $y_{AK} = \omega_0 v + 2 \sum_{\substack{j=1 \\ j \text{ odd}}}^{\nu-1} \Re(\omega_j x_j)$.

As already mentioned, the solution of $Mu_R = \widehat{w}$ is performed iteratively, as M should not be explicitly computed but only applied in products such as $y = Mx$, as is the case in conjugate gradient methods. Each matrix-vector multiplication with M requires solving two systems with $B = \tau I - A$, and this is related to the fact that M is the coefficient matrix of the normal equation.

The final attainable accuracy of the overall computation depends primarily on the rational function used, but also on the accuracy with which the linear systems of step (iii) are solved. This requires the stopping tolerance to be smaller than the accuracy requested; in our experiments we noticed that $tol = 10^{-\nu-2}$ delivered a sufficiently accurate final solution to the exponential. No further study was attempted to refine this value.

We consider solving systems with B with both a direct and an iterative method. In the former case, the cost of factorizing the single matrix B is performed once for all systems. This provides significant computational savings over the original method in [1], a sample of which is reported in Table 4.2. In the table we compare the original method, where an optimal $B = B(\tau)$ is determined and factorized for each pole, with Algorithm AK, where a single suboptimal B is computed and factorized at step (ii) of the algorithm. The numbers show that the new strategy improves performance,

TABLE 4.2
Comparison of the original method in [1] *and Algorithm AK for Example* 5.1 *in section* 5. *Direct methods are used to solve the linear systems.*

n	*tol*	Original method time (# its)	Algorithm AK time (# its)
125	10^{-5}	0.02 (10)	0.02 (11)
	10^{-8}	0.04 (23)	0.23 (32)
	10^{-11}	0.05 (40)	0.05 (61)
	10^{-14}	0.07 (69)	0.09(105)
3375	10^{-5}	1.59 (11)	1.31 (11)
	10^{-8}	2.92 (25)	2.23 (28)
	10^{-11}	4.80 (46)	4.08 (53)
	10^{-14}	6.87 (72)	5.97 (85)
15625	10^{-5}	30.42 (11)	22.14 (11)
	10^{-8}	55.08 (25)	31.77 (27)
	10^{-11}	84.32 (46)	54.90 (51)
	10^{-14}	119.97 (74)	77.73 (84)

TABLE 4.3
Problems of section 5. *CPU time of Algorithm AK when different iterative schemes are used to solve with* $B = \tau I - A$.

n	*tol*	AK+Variant	AK+CG	AK+PCG
Example 5.1				
125	10^{-5}	0.02	0.04	0.05
	10^{-8}	0.04	0.07	0.08
	10^{-11}	0.08	0.15	0.17
	10^{-14}	0.15	0.29	0.32
3375	10^{-5}	0.42	0.65	1.22
	10^{-8}	0.77	1.75	2.91
	10^{-11}	1.73	3.88	6.07
	10^{-14}	2.81	6.69	11.11
15625	10^{-5}	3.20	4.57	8.61
	10^{-8}	5.88	13.31	21.21
	10^{-11}	13.42	28.07	44.51
	10^{-14}	22.10	52.51	83.22
Example 5.2				
2500	10^{-5}	0.68	1.38	1.10
	10^{-8}	1.69	4.02	3.01
	10^{-11}	3.43	8.32	8.46
	10^{-14}	5.86	15.70	12.58
10000	10^{-5}	3.67	9.38	7.69
	10^{-8}	8.60	28.54	22.34
	10^{-11}	17.50	61.85	47.12
	10^{-14}	29.54	122.99	89.27

especially on large problems, while the total number of iterations does not significantly grow, compared to the optimal shift selection in [1]. The results in the table were obtained by using a direct method for solving the involved inner linear systems. Additional numerical experiments, not reported here, with iterative solves confirmed the superiority of our new approach even in this inner-outer setting.

In the case an iterative solver is used, one is faced with the problem of efficiently solving two systems with B at each iteration of the solver with M. By exploiting the positive definiteness of B, we consider the following alternatives: (a) two calls to the conjugate gradients in sequence; (b) two calls to preconditioned CGs in sequence; (c) one call to a variant of the CG method proposed by van der Vorst in [43] to

simultaneously solve for B and B^2 with a single recurrence. The CPU times obtained for the problems of Examples 5.1 and 5.2 are reported in Table 4.3.

The numbers show that the variant that simultaneously approximates the systems with B and B^2 is faster than both the standard CG method and its preconditioned version. It is important to notice that in the approach proposed in [43] preconditioning is not applicable; nonetheless, its performance is superior to that of standard preconditioned CG applied twice. We should mention that when using the approach in [43], one could employ a different (optimal) B for each shifted system at no additional cost. We decided to maintain Algorithm AK for consistency with the case of the direct solves.

5. Numerical experiments. In this section we report on our numerical experience with the discussed methods, which we summarize as follows:

- (PFE). Computation of (2.1) by explicitly solving each complex symmetric system. Systems corresponding to conjugate pairs are coupled.
- . Classical Lanczos approach described in section 2; see, e.g., [34].
- (AK). Variant of method by Axelsson and Kucherov described in section 4.
- (SI). Acceleration procedure described in section 3 [41].

When using methods that explicitly rely on the partial fraction expansion, namely PFE and AK, the final accuracy influences the number of terms in the expansion, and thus the number of shifted systems to be solved. In our implementation of the shift-invert procedure, the parameter selection is also guided by the required accuracy; cf. Table 3.3. In general, this is not strictly necessary, and one could choose σ as the optimal parameter associated with an approximation of large degree, say $\nu = 14$.

Since the error norm cannot be explicitly monitored, stopping criteria were introduced as discussed in previous sections. In the small examples, however, we computed the actual error and verified that a satisfactory tolerance was reached, achieving the required order of magnitude for the absolute error norm. It should be mentioned that the solution norm influences the stopping criterion. Indeed, depending on the spectrum of A, the vector $\exp(A)v$ may have a very small norm, which makes a loose stopping tolerance completely useless, yielding an approximate solution with no digit accuracy. In all problems considered, the vector $\exp(tA)v$ with $t = 0.1$ had a norm not smaller than 10^{-4}.

All methods except the standard Lanczos procedure require solving real or complex shifted systems. In all of these cases, such a step employs over 95% of the total computational efforts, so that it is really the only bottleneck of these methods. In the next two subsections we report results when solving these systems by either a direct or an iterative method, yielding in this latter case an inner-outer procedure.

All numerical experiments of this section were performed in MATLAB [28], version 7.0.1 (R14-SP1), and CPU timings were obtained with the function `cputime`. We like to mention that different CPU time performance was observed when using different MATLAB versions or releases, which in some cases significantly affected the comparison among the methods.

5.1. Using direct methods. In this section, we report our experiments when a direct method is used to solve with $(A-\xi_j I)$ or $(\tau I - A)$. When dealing with the symmetric and positive definite matrix $(\tau I - A)$, the Cholesky factorization is performed, after a reordering of the matrix entries (MATLAB function `symamd`). Permutation sig-

nificantly improves the overall cost of solving with the shifted matrix (both the factorization and the solution phases). In the following, the matrix is always reordered, and the reported timings include the factorization cost. The LU decomposition without pivoting the complex symmetric matrix $(A - \xi_j I)$ yields a (symmetric) LDL^T factorization. After reordering, the number of (now complex) nonzero entries is the same as for the real factor. We emphasize, however, that in the case of the PFE, the complex system solutions were carried out by means of the MATLAB backslash operation "\", which is significantly faster than the two-step procedure of first factorizing the matrix and then solving with the factors.

In all tables, the number of iterations for standard Lanczos and for SI coincides with the dimension of the generated Krylov subspace. For AK, the number in parentheses is the global number of iterations performed to solve all shifted systems with $M = M(\xi_j)$. The stopping tolerance $tol = 10^{-\nu}$ is fixed for all codes. Methods employing the rational function approximation thus use the corresponding function degree ν.

5.1. We consider the $n \times n$ matrix stemming from the finite difference discretization of the 3D Laplace operator on the unit cube and Dirichlet homogeneous boundary conditions, with eigenvalues in $[-179.14, -12.862]$ for $n = 125$. Different discretization refinements are considered. These data represent a typical benchmark for the approximate evaluation of the matrix exponential in PDE contexts. We approximate the vector $\exp(tA)v$, with $t = 0.1$ and v a normalized normally distributed random vector. The elapsed time and the number of iterations (in parentheses) for various problem dimensions and final tolerances are reported in Table 5.1.

TABLE 5.1

Example 5.1. *CPU time (and number of iterations in parentheses when appropriate) for all methods when systems with shifted matrices are solved with a direct method. Different dimension problems and various stopping tolerances are reported.*

n	tol	Standard Lanczos	PFE	AK	SI
125	10^{-5}	0.01 (13)	0.01	0.02 (11)	0.01 (7)
	10^{-8}	0.01 (18)	0.01	0.03 (32)	0.01 (11)
	10^{-11}	0.01 (22)	0.03	0.05 (61)	0.01 (14)
	10^{-14}	0.01 (24)	0.03	0.08(105)	0.01 (17)
3375	10^{-5}	0.14 (47)	1.32	1.33 (11)	0.48 (8)
	10^{-8}	0.21 (55)	2.13	2.23 (28)	0.65 (13)
	10^{-11}	0.35 (67)	2.88	4.07 (53)	0.85 (19)
	10^{-14}	0.52 (77)	3.70	5.94 (85)	1.06 (25)
15625	10^{-5}	2.69 (89)	30.35	22.05 (11)	11.49 (10)
	10^{-8}	2.95 (93)	51.61	31.60 (27)	11.88 (11)
	10^{-11}	4.76 (113)	69.03	54.68 (51)	14.22 (17)
	10^{-14}	7.25 (130)	90.20	77.31 (84)	16.96 (24)

Several comments are in order. First, we observe that explicitly dealing with the PFE by means of direct solvers becomes significantly more expensive, especially for the large size matrix. Moreover, all methods behave quite consistently as the problem dimension increases, and the performance ranking is already clear for $n = 3375$.

Second, on this problem, standard Lanczos is the most efficient approach, as far as CPU time is concerned; the shift-invert procedure shows the second best performance. Memory requirements for standard Lanczos, however, become increasingly high as the number of iterations increases, since the whole basis needs to be stored. This problem may be overcome by resorting to a two-pass strategy. In the first pass, the Lanczos

basis is not stored, but only the projected solution is; in the second pass, the Lanczos basis vectors are recomputed one at a time to form the final solution; see [19] for more details. This approach drastically reduces memory needs, but requires almost twice the time to complete the computation, making the performance of the method more comparable to that of SI. For the latter method, we observe that the number of iterations does not grow (in fact, it decreases) as the problem dimension increases for the same required tolerance; see, e.g., [30], [41] for a discussion.

Third, the AK approach does not perform satisfactorily, as compared to the Lanczos methods, although its memory requirements are roughly limited to the Cholesky factor and to a few CG vectors. The number of iterations does not grow significantly as the problem dimension increases for a fixed final tolerance. We should mention that AK improves the performance of the original PFE method on the large matrix. This seems to indicate that AK may be advantageous for approximating the action of other matrix rational functions for which the standard Lanczos procedure does not show superlinear convergence. Moreover, the method's limitations are less apparent when a loose final accuracy is required, which is precisely the context suggested in the original paper [1].

In summary, this example shows that for moderately large spectra, the standard Lanczos approach is still competitive, and the analyzed acceleration procedures do not seem to significantly improve its performance. The next example faces a more extreme case for which using an acceleration procedure is mandatory.

EXAMPLE 5.2. In this example we approximate $\exp(tA)v$, $t = 0.1$, where the $n \times n$ matrix A stems from the finite difference discretization of the 2D operator

$$\mathcal{L}(u) = (a(x,y)u_x)_x + (b(x,y)u_y)_y, \qquad a(x,y) = 1 + y - x, \quad b(x,y) = 1 + x + x^2$$

on the unit square, with Dirichlet homogeneous boundary conditions [41]. Two grid refinements have been considered. The spectrum is contained in the interval $[-35424, -25.256]$ for the smaller problem. The vector v is taken as in the previous example. The CPU time and number of iterations, when appropriate, are reported in Table 5.2. This example has special features that make it very different compared to the previous one. In particular, $\|A\|$ and the spectral range are significantly large, penalizing the standard Lanczos method. Moreover, the finite difference discretization of the 2D operator generates a sparser matrix than in Example 5.1, allowing cheaper system solves. We can thus predict especially good performance of all acceleration techniques, including PFE, compared to standard Lanczos. The results in Table 5.2 fully confirm these considerations.

5.2. Using iterative methods. The use of iterative methods for solving the large linear systems provides a significantly different picture from what is shown in the previous section. In the case of AK and SI, the resulting algorithm is an inner-outer procedure. We next compare the standard Lanczos method with the following iterative procedures:

- PFE+QMR. Partial fraction expansion where each complex shifted system is solved by a preconditioned simplified QMR method [18]. The preconditioner is a complex symmetric LDL^T incomplete factorization of the shifted matrix, obtained by a simple modification of the factors computed with the MATLAB `luinc` factorization with dropping tolerance equal to 10^{-2}. The system stopping threshold is $10^{-\nu}$.
- SI+PCG. Shift-invert Lanczos where systems with $I - \sigma A$ are solved with preconditioned conjugate gradients. The MATLAB `cholinc` function with

TABLE 5.2

Example 5.2. *CPU time (and the number of iterations in parentheses when appropriate) for all methods when systems with shifted matrices are solved with a direct method. Different dimension problems and various stopping tolerances are reported.*

n	*tol*	Standard Lanczos	PFE	AK	SI
2500	10^{-5}	16 (194)	0.22	0.29 (11)	0.12 (10)
	10^{-8}	18 (200)	0.33	0.50 (27)	0.13 (11)
	10^{-11}	53 (242)	0.44	0.92 (51)	0.20 (19)
	10^{-14}	111 (280)	0.53	1.39 (84)	0.24 (24)
10000	10^{-5}	615 (406)	1.24	1.39 (9)	0.67 (11)
	10^{-8}	610 (406)	1.87	2.53 (25)	0.66 (11)
	10^{-11}	1221 (484)	2.55	4.71 (47)	0.94 (17)
	10^{-14}	- (> 500)	3.20	7.49 (82)	1.24 (23)

dropping tolerance 10^{-2} is used to generate the preconditioner. The inner system stopping threshold is $10^{-\nu}$.

- AK+Variant. We report the results of Table 4.3 of the variant of the Axelsson–Kucherov method, which solves systems with $B = \tau I - A$ and B^2 with a single iterative method. If occurring, the system with the real pole is solved with preconditioned conjugate gradients as in SI+PCG. The inner system stopping threshold is $10^{-\nu-3}$.

In SI+PCG and AK+Variant, the shifted matrix was reordered with a Cuthill–McKee permutation (MATLAB function `symrcm`) before building the preconditioner, whereas minimum degree reordering was used for PFE+QMR; see [5] for a comprehensive discussion of various permutations related to preconditioning. We should mention that, in SI, it is not necessary to solve the inner system at high accuracy but that, on the contrary, the accuracy can be relaxed as convergence takes place [41], [38]. We postpone the exploration of this alternative to the next section, where enhancement strategies for the PFE+QMR algorithm are also devised.

The CPU times for the two test problems are reported in Table 5.3 (the results for $n = 125$ are omitted). For SI+PCG and AK+Variant, the total number of outer iterations and the average number of inner iterations are shown. For PFE+QMR, the average number of iterations is also shown in parentheses. For ease of comparison, we also reproduce the CPU time of standard Lanczos from Tables 5.1–5.2.

Compared to the previous results that used a direct solver for the shifted systems, we can see that the overall costs have significantly decreased for all methods except SI. In the case of the 3D Laplace operator (Example 5.1), the standard Lanczos method remains the method of choice even after a two-step procedure, although the differences are far less prominent. For the 2D operator the solution of the shifted systems in (1.1) with an appropriately preconditioned iterative solver yields the most competitive approach, even for the small size problem. It appears that, for these examples, the two preconditioners $(A - \tau I)$, corresponding to SI and the incomplete LDL^T factorization for PFE, show comparable performance in approximating the PFE, in spite of stemming from rather different approximation strategies.

6. Further tests. In this section we explore performance enhancements for the two methods SI+PCG and PFE+QMR. All experiments in this section were carried out on one processor of a Sun Fire V40z with 2390.895 MHz and 16 GB RAM, running MATLAB 7.4. We first discuss some natural implementation improvements for both algorithms and then analyze their performance on a time-stepping problem, so as to

TABLE 5.3
Approximation when shifted systems are solved with iterative methods.

n	*tol*	Standard Lanczos	PFE+ QMR (avg its)	SI+ PCG (out/avg in)	AK+ Variant (out/avg in)
			Example 5.1		
3375	10^{-5}	0.14	0.67 (8)	0.44 (8/7)	0.42 (20/6)
	10^{-8}	0.21	1.15 (11)	0.81 (13/9)	0.77 (30/7)
	10^{-11}	0.35	1.75 (14)	1.27 (19/10)	1.73 (69/11)
	10^{-14}	0.52	2.30 (16)	1.94 (25/12)	2.81(89/126)
15625	10^{-5}	2.69	5.29 (11)	4.05 (10/10)	3.20 (23/7)
	10^{-8}	2.95	9.36 (17)	5.37 (11/13)	5.88 (29/7)
	10^{-11}	4.76	14.29 (22)	8.87 (17/15)	13.42 (74/12)
	10^{-14}	7.25	19.52 (27)	14.39 (24/18)	22.10 (86/12)
			Example 5.2		
2500	10^{-5}	16	0.36 (13)	0.54 (10/12)	0.68 (25/8)
	10^{-8}	18	0.68 (18)	0.75 (11/16)	1.69 (29/7)
	10^{-11}	53	1.09 (22)	1.46 (19/18)	3.43 (76/13)
	10^{-14}	111	1.54 (26)	2.12 (24/21)	5.86 (87/12)
10000	10^{-5}	615	2.46 (24)	4.4 (11/21)	3.6 (32/10)
	10^{-8}	610	4.92 (35)	5.5 (11/27)	8.6 (27/ 7)
	10^{-11}	1221	8.17 (43)	9.8 (17/32)	17.5 (92/15)
	10^{-14}	-	11.74 (51)	15.4 (13/37)	29.5 (95/13)

provide a more realistic framework.

As already mentioned, the inner-outer version of SI may be implemented so as to relax the accuracy with which the inner system is solved at each Lanczos iteration. Given a fixed tolerance $\epsilon > 0$, in [41, (5.4)] the following stopping tolerance η_j for the inner system was proposed:

$$\eta_j = \frac{\epsilon}{\|e_{j-1}\| + \epsilon},$$

where e_{j-1} is the error in the approximation of the exponential operator at the previous iteration, $j-1$. In practice, $\|e_{j-1}\|$ is replaced by an estimate; in our implementation we used the estimate associated with (4.9) in [41], and we fixed ϵ to be equal to the initial inner tolerance. Clearly, η_j increases towards one as $\|e_{j-1}\| \to 0$, so that the inner solver may be stopped earlier as the outer iteration converges, thus hopefully decreasing the overall computational costs. We refer to [38], [42] for a general discussion on relaxation strategies.

A straightforward enhancement for the partial fraction evaluation (hereafter PFE+QMR+mono) is to compute a single preconditioner, and then to apply it to all systems. This strategy makes the SI+PCG and the PFE+QMR methods even closer to each other, since we have shown that SI+PCG may be viewed as a special way of preconditioning the PFE systems with a single, parameter dependent matrix. Here, we take as a single preconditioner of PFE+QMR+mono the factor of the incomplete Cholesky factorization (MATLAB 7.4 function `cholinc`) with dropping tolerance 10^{-2} of the shifted complex symmetric matrix $A - \xi_1 I$, where ξ_1 is the pole with the largest imaginary part, which provided the best performance. Reordering with `symamd` of the matrix A (and of I) was performed before calling `cholinc`. The results obtained for the problem of Example 5.2 are reported in Table 6.1. The original methods PFE+QMR and SI+PCG, together with their enhanced versions, PFE+QMR+mono and SI+PCG+relax, are displayed. For SI+PCG+relax, the initial inner tolerance was

TABLE 6.1

Example 5.2. *CPU time and the number of iterations for the original PFE+QMR and SI+PCG methods and for their enhanced versions.*

n	*tol*	PFE+QMR mono (avg its)	PFE+QMR (avg its)	SI+PCG relax (out/avg in)	SI+PCG (out/avg in)
2500	10^{-5}	0.20 (12)	0.18 (12)	**0.15** (9/ 7)	0.29 (10/11)
	10^{-8}	**0.27** (16)	0.32 (16)	**0.27** (15/ 9)	0.32 (11/15)
	10^{-11}	0.47 (20)	0.53 (20)	**0.42** (21/10)	0.61 (19/17)
	10^{-14}	0.63 (24)	0.73 (23)	**0.58** (26/12)	0.90 (24/21)
10000	10^{-5}	1.31 (22)	1.37 (22)	**1.18** (8/14)	2.50 (11/21)
	10^{-8}	**2.19** (31)	2.67 (32)	2.22 (14/16)	3.05 (11/27)
	10^{-11}	3.99 (39)	4.32 (39)	**3.65** (20/18)	5.44 (17/31)
	10^{-14}	**5.29** (45)	6.14 (46)	5.33 (26/20)	8.49 (23/37)

TABLE 6.2

Parabolic problem (cf. (6.1)*). CPU times of standard Lanczos and of enhanced accelerated methods to approximate the solution at* $\mathbf{T} = 0.1$*, for different time step lengths* δt *and different number of nodes* nx, ny *in the discretization of the domain* $(0,1)^2$*.*

Grid (nx, ny)	Final accuracy	δt	Standard Lanczos	SI+PCG relax	PFE+QMR mono
(50,50)	10^{-4}	5.0e-03	0.28	0.76	1.59
		1.0e-02	0.30	0.54	0.89
		5.0e-02	1.17	0.24	0.27
		1.0e-01	2.66	0.17	0.15
	10^{-6}	5.0e-03	0.32	1.01	2.99
		1.0e-02	0.42	0.70	1.59
		5.0e-02	2.43	0.36	0.42
		1.0e-01	6.61	0.27	0.24
(90,90)	10^{-4}	5.0e-03	1.60	5.23	7.92
		1.0e-02	2.25	3.73	5.03
		5.0e-02	18.08	1.74	1.58
		1.0e-01	64.14	1.28	0.93
	10^{-6}	5.0e-03	2.30	6.89	14.71
		1.0e-02	3.59	4.96	8.75
		5.0e-02	50.03	2.58	2.61
		1.0e-01	181.16	1.91	1.54
(120,120)	10^{-4}	5.0e-03	5.07	12.58	19.82
		1.0e-02	9.08	8.76	13.00
		5.0e-02	231.26	4.02	3.76
		1.0e-01	883.69	3.10	2.10
	10^{-6}	5.0e-03	6.75	16.69	35.06
		1.0e-02	11.84	12.15	21.42
		5.0e-02	258.84	6.08	6.50
		1.0e-01	883.32	4.79	3.67

equal to the outer tolerance. The improvement over the corresponding original method is significant, reaching almost 50% for SI+PCG+relax in some instances. Note that, on this problem, the single preconditioned PFE+QMR+mono is very effective for all systems, allowing for the same average number of iterations as of the original method. In bold are the best timings, which show that the two different enhanced preconditioned techniques, PFE+QMR+mono and SI+PCG+relax, behave quite similarly. In general, timings are so close, the difference being within the MATLAB timings fluctuation that it is difficult to depict a clear winner.

We next consider the discretization of the following parabolic equation in two spatial dimensions [41]:

$$\frac{\partial}{\partial t}u = \mathcal{L}(u), \quad (x,y) \in (0,1)^2, \quad 0 \le t \le \mathbf{T}, \tag{6.1}$$

where the solution $u = u(t,x,y)$ is subject to the initial condition $u(0,x,y) = u_0(x,y)$ and to mixed boundary conditions (b.c.): homogeneous Dirichlet b.c. on the western and eastern boundaries, and homogeneous Neumann b.c. on the northern and southern boundaries of the domain. The operator $\mathcal{L}$ is as in Example 5.2. After standard centered finite difference space discretization, the solution at $t = \mathbf{T}$ is approximated by a sequence of $\mathbf{T}/\delta t$ applications of $\exp(\delta t A)$ as $\exp(\delta t A) \cdots \exp(\delta t A) u_0$, where δt is the time step length and u_0 is the initial vector. We considered different possible space and time discretizations so as to approximate the exact solution at $\mathbf{T} = 0.1$. The results of our experiments for $\delta t = 0.005, 0.01, 0.05, 0.1$ are reported in Table 6.2; u_0 is the normalized vector of all ones. The standard Lanczos and the enhanced versions of the SI and PFE methods are considered. Different final accuracies were also used, which are of interest in the context of evolution problems. All inner and outer stopping thresholds were tuned so as to reach the requested final accuracy.

The acceleration procedures allow the discretization process to take much larger time steps than with standard Lanczos, to the point that in all examples a single time step ($\delta t = 0.1$) is faster than the best Lanczos timing. This is clearly a welcome event and is one of the main reasons for using acceleration procedures in the context of parabolic equations. In addition, we explicitly observe that as the number of time steps decreases, so does the cost of the acceleration procedures, whereas that of standard Lanczos becomes unacceptably high due to the increasing value of $\|\delta t A\|$.

In PFE, the common preconditioner is computed once and for all, whereas each shifted system is solved separately. This is the major remaining drawback of the enhanced PFE+QMR method when a few time steps are performed, since many systems need to be solved. On the other hand, SI precisely avoids this step, since it constructs a single preconditioner that, in the case of a rational function, still allows one to keep the shifted form of the systems, so that all systems can be solved simultaneously with a single SI iteration as in (3.6); see also [19]. Albeit limited, our numerical experiments confirm that the relaxed SI method is able to efficiently solve the parabolic system when few time steps are taken, compared to standard Lanczos. As already noticed in the previous example, a single time step makes the generic enhanced PFE procedure more competitive. Due to these favorable results of the PFE+QMR method, it would be interesting to further explore enhancement techniques for this approach, such as those in [7].

7. Conclusions. In this paper we have presented a common framework for some recently developed acceleration techniques for approximating the action of the matrix exponential to a vector. This framework is based on the rational function approximation to the exponential, which allows one to transform the approximation problem into that of solving several algebraic linear systems. It is thus natural to compare the performance of the acceleration techniques with that of methods such as PFE that explicitly solve these systems. We can summarize our theoretical and experimental findings as follows:

(i) Whenever the exponential is replaced by its rational function approximation, we have shown that the analyzed methods SI and AK are simply different ways of preconditioning the given linear systems.

(ii) Our framework allowed us to derive an a priori fully algebraic and iteration-free selection of the involved single parameter in both methods.

(iii) We have performed a numerical comparison among various acceleration methods, including their enhanced versions, thus filling a gap in the current literature. In our opinion, these experiments provide a new perspective and new insights on when and which acceleration procedures should be preferred. The experiment on a parabolic 2D problem shows the effectiveness of the enhanced SI and PFE+QMR processes, allowing the time discretization for truly large iteration steps.

Our findings are in fact quite general. It would be interesting to see whether similar conclusions can be generalized to other functions for which a rational function approximation is available; see, e.g., [19]. The case of nonsymmetric A is also very challenging, since the rational Chebyshev approximation is not optimal in this case.

Acknowledgments. The authors would like to thank Marlis Hochbruck and Igor Moret for insightful conversations, and the two anonymous referees for their constructive criticism, which led to the addition of section 6. Finally, we are grateful to Andreas Frommer for his handling of the manuscript.

REFERENCES

[1] O. Axelsson and A. Kucherov, *Real valued iterative methods for solving complex symmetric linear systems*, Numer. Linear Algebra Appl., 7 (2000), pp. 197–218.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. A. van der Vorst, eds., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.

[3] G. A. Baker and P. Graves-Morris, *Padé Approximants*, Encyclopedia Math. Appl. 59, 2nd ed., Cambridge University Press, Cambridge, UK, 1996.

[4] C. Baldwin, R. Freund, and E. Gallopoulos, *A parallel iterative method for exponential propagation*, in Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing, D. Bailey et al., eds., SIAM, Philadelphia, 1995, pp. 534–539.

[5] M. Benzi, *Preconditioning techniques for large linear systems: A survey*, J. Comput. Phys., 182 (2002), pp. 418–477.

[6] M. Benzi and D. Bertaccini, *Approximate inverse preconditioning for shifted linear systems*, BIT, 43 (2003), pp. 231–244.

[7] M. Benzi and D. Bertaccini, *Block Preconditioning of Real-valued Iterative Algorithms for Complex Linear Systems*, to appear in IMA J. Numer. Anal.

[8] L. Bergamaschi and M. Vianello, *Efficient computation of the exponential operator for large, sparse, symmetric matrices*, Numer. Linear Algebra Appl., 7 (2000), pp. 27–45.

[9] A. J. Carpenter, A. Ruttan, and R. S. Varga, *Extended numerical computations on the* 1/9 *conjecture in rational approximation theory*, in Rational Approximation and Interpolation, Lecture Notes in Math. 1105, P. R. Graves-Morris, E. B. Saff, and R. S. Varga, eds., Springer-Verlag, Berlin, 1984, pp. 383–411.

[10] P. Castillo and Y. Saad, *Preconditioning the matrix exponential operator with applications*, J. Sci. Comput., 13 (1999), pp. 275–302.

[11] W. J. Cody, G. Meinardus, and R. S. Varga, *Chebyshev rational approximations to* e^{-x} *in* $[0, +\infty)$ *and applications to heat-conduction problems*, J. Approx. Theory, 2 (1969), pp. 50–65.

[12] V. Druskin, A. Greenbaum, and L. Knizhnerman, *Using nonorthogonal Lanczos vectors in the computation of matrix functions*, SIAM J. Sci. Comput., 19 (1998), pp. 38–54.

[13] V. Druskin and L. Knizhnerman, *Two polynomial methods of calculating functions of symmetric matrices*, USSR Comput. Math. Math. Phys., 29 (1989), pp. 112–121.

[14] V. Druskin and L. Knizhnerman, *Krylov subspace approximation of eigenpairs and matrix functions in exact and computer arithmetic*, Numer. Linear Algebra Appl., 2 (1995), pp. 205–217.

[15] V. Druskin and L. Knizhnerman, *Extended Krylov subspaces: Approximation of the matrix square root and related functions*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 755–771.

[16] M. Eiermann and O. Ernst, *A restarted Krylov subspace method for the evaluation of matrix functions*, SIAM J. Numer. Anal., 44 (2006), pp. 2481–2504.

[17] R. W. Freund, *On conjugate gradient type methods and polynomial preconditioners for a class of complex non-Hermitian matrices*, Numer. Math., 57 (1990), pp. 285–312.
[18] R. W. Freund and N. M. Nachtigal, *Software for simplified Lanczos and QMR algorithms*, Appl. Numer. Math., 19 (1995), pp. 319–341.
[19] A. Frommer and V. Simoncini, *Stopping criteria for rational matrix functions of Hermitian and symmetric matrices*, SIAM J. Sci. Comput., 30 (2008), pp. 1387–1412
[20] E. Gallopoulos and Y. Saad, *Efficient solution of parabolic equations by Krylov approximation methods*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1236–1264.
[21] G. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.
[22] N. J. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1179–1193.
[23] M. Hochbruck and M. E. Hochstenbach, *Subspace Extraction for Matrix Functions*, preprint, 2005.
[24] M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925.
[25] M. Hochbruck and C. Lubich, *Exponential integrators for quantum-classical molecular dynamics*, BIT, 39 (1999), pp. 620–645.
[26] A. Iserles, *How large is the exponential of a banded matrix?*, New Zealand J. Math., 29 (2000), pp. 177–192.
[27] L. Lopez and V. Simoncini, *Analysis of projection methods for rational function approximation to the matrix exponential*, SIAM J. Numer. Anal., 44 (2006), pp. 613–635.
[28] The MathWorks, Inc., *MATLAB* 7, 2004.
[29] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Rev., 45 (2003), pp. 3–49.
[30] I. Moret, *On RD-rational Krylov approximations to the core-functions of exponential integrators*, Numer. Linear Algebra Appl., 14 (2007), pp. 445–457.
[31] I. Moret and P. Novati, *An interpolatory approximation of the matrix exponential based on Faber polynomials*, J. Comput. Appl. Math., 131 (2001), pp. 361–380.
[32] I. Moret and P. Novati, *RD-rational approximations of the matrix exponential*, BIT, 44 (2004), pp. 595–615.
[33] I. Moret and P. Novati, *Interpolating functions of matrices on zeros of quasi-kernel polynomials*, Numer. Linear Algebra Appl., 11 (2005), pp. 337–353.
[34] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 29 (1992), pp. 209–228.
[35] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[36] E. B. Saff, A. Schönhage, and R. S. Varga, *Geometric convergence to e^{-z} by rational functions with real poles*, Numer. Math., 25 (1976), pp. 307–322.
[37] R. B. Sidje, *Expokit: A Software Package for Computing Matrix Exponentials*, ACM Trans. Math. Softw., 24 (1998), pp. 130–156.
[38] V. Simoncini and D. B. Szyld, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM J. Sci. Comput., 25 (2003), pp. 454–477.
[39] V. Simoncini and D. B. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59.
[40] L. N. Trefethen, J. A. C. Weideman, and T. Schmelzer, *Talbot quadratures and rational approximations*, BIT, 46 (2006), pp. 653–670.
[41] J. van den Eshof and M. Hochbruck, *Preconditioning Lanczos approximations to the matrix exponential*, SIAM J. Sci. Comput., 27 (2006), pp. 1438–1457.
[42] J. van den Eshof and G. L. G. Sleijpen, *Inexact Krylov subspace methods for linear systems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 125–153.
[43] H. A. van der Vorst, *An iterative solution method for solving $f(A)x = b$ using Krylov subspace information obtained for the symmetric positive definite matrix A*, J. Comput. Appl. Math., 18 (1987), pp. 249–263.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 684–699

DEFLATION AND BALANCING PRECONDITIONERS FOR KRYLOV SUBSPACE METHODS APPLIED TO NONSYMMETRIC MATRICES*

YOGI A. ERLANGGA† AND REINHARD NABBEN†

Abstract. For quite some time, the deflation preconditioner has been proposed and used to accelerate the convergence of Krylov subspace methods. For symmetric positive definite linear systems, the convergence of conjugate gradient methods combined with deflation has been analyzed and compared with other preconditioners, e.g., with the abstract balancing preconditioner [R. Nabben and C. Vuik, *SIAM J. Sci. Comput.*, 27 (2006), pp. 1742–1759]. In this paper, we extend the convergence analysis to nonsymmetric linear systems in the context of GMRES iteration and compare it with the abstract nonsymmetric balancing preconditioner. We are able to show that many results for symmetric positive definite matrices carry over to arbitrary nonsymmetric matrices. First we establish that the spectra of the preconditioned systems are similar. Moreover, we show that under certain conditions, the 2-norm of residuals produced by GMRES combined with deflation is never larger than the 2-norm of residuals produced by GMRES combined with the abstract balancing preconditioner. Numerical experiments are done to nonsymmetric linear systems arising from a finite volume discretization of the convection-diffusion equation, and the numerical results confirm our theoretical results.

Key words. deflation, balancing preconditioner, nonsymmetric matrix, GMRES, convection-diffusion

AMS subject classifications. 65F10, 65F50, 65N22, 65N55

DOI. 10.1137/060678257

1. Introduction.

For a linear system

$$Au = b, \qquad A \in \mathbb{R}^{n\times n}, \tag{1.1}$$

where A is a large but sparse nonsymmetric, nonsingular matrix, GMRES [23], among others, is a popular method to iteratively solve it. Such a system is encountered, for example, when a discretization is applied to the steady convection-diffusion equation. For starting vector u_0, GMRES constructs a sequence of vectors (called Arnoldi vectors) using Arnoldi orthogonalization [2], which forms the basis for the Krylov subspace, i.e., the subspace

$$\mathcal{K}^k(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0\}, \quad r_0 = b - Au_0. \tag{1.2}$$

The approximate solution at the kth iteration, denoted by u_k, is then contained in the affine subspace $u_0 + \mathcal{K}^k(A, r_0)$, i.e.,

$$u_k \in u_0 + \mathcal{K}^k(A, r_0). \tag{1.3}$$

In case of GMRES, u_k minimizes the 2-norm of the residual over the subspace.

In many applications, however, GMRES exhibits slow convergence. Since all Arnoldi vectors are needed during orthogonalization, slow convergence increases the

*Received by the editors December 21, 2006; accepted for publication (in revised form) by M. Benzi February 12, 2008; published electronically June 18, 2008. This research was supported by the *Deutsche Forschungsgemeinschaft* (DFG), project NA248/2-2.
http://www.siam.org/journals/simax/30-2/67825.html
†Institut für Mathematik, TU Berlin, MA 3-3, Strasse des 17. Juni 136, D-10623 Berlin, Germany (erlangga@math.tu-berlin.de, nabben@math.tu-berlin.de).

number of Arnoldi vectors being used and stored in the computer memory. This makes GMRES often impractical. A simple remedy to the memory requirement is by restarting GMRES after j iterations, as already suggested in [23], denoted by GMRES(j) throughout. The suitable value of j, the restarting parameter, is generally not known, and an inappropriate value of j may lead GMRES to stagnation.

Morgan [15] proposed a remedy in the context of GMRES(j) by reusing information already obtained during the first j iterations. Vectors related to the converged eigenvectors available during the first j GMRES iterations are added to the subspace before restarting; thus, the subspace is [illegible]. Adding these vectors removes (or [illegible]) the corresponding (small) eigenvalues from the spectrum. Related work can also be found in [11, 9, 4]. See also [8] for a unified overview on this class of methods. Recently, these deflation techniques have been successfully used in the numerical solution of several practical problems; see, e.g., [6, 1, 5, 20, 12].

A similar idea has also been used in the context of preconditioned conjugate gradient (CG) methods for symmetric positive definite (spd) systems; see, e.g., [18, 10]. As the convergence of CG is related to the condition number of the spd linear system to solve, deflation is used to improve the condition number by shifting some of the smallest eigenvalues to zero. Since the corresponding eigenvectors no longer have components during the iterations [18], CG will converge faster. Here, we can speak of the [illegible] condition number after deflation, which is never larger than the [illegible] condition number. In [16, 17], the deflation-based preconditioner is analyzed and compared with the [illegible] form of the coarse-grid correction preconditioner [19] and the [illegible] form of the balancing preconditioner [13]. In theory, the deflation vectors are not necessarily invariant vectors and, more generally, can also be related to the prolongation matrix in the multigrid language.

It is somewhat worthwhile to extend the analysis to nonsymmetric systems. This is the aim of this paper. In addition, we compare deflation with the abstract balancing preconditioner as well. For this purpose, we define the abstract deflation preconditioners as

$$P_D = I - AZE^{-1}Y^T, \quad Q_D = I - ZE^{-1}Y^TA, \quad E = Y^TAZ, \tag{1.4}$$

where P_D and Q_D are related to the left and right preconditioners, respectively. One can easily show that P_D and Q_D are projectors, i.e., $P_D^2 = P_D$ and $Q_D^2 = Q_D$. Here, Z and Y are suitable deflation subspaces of dimension $n \times r$, where $r \ll n$, and hence E is presumably easy to compute and invert. Note that Z and Y are arbitrary matrices. We do not assume that their columns are eigenvectors or approximations of eigenvectors.

In deflation, the solution of (1.1) is computed as follows. We decompose the solution u into

$$u = (I - Q_D)u + Q_Du = ZE^{-1}Y^Tb + Q_Du. \tag{1.5}$$

As the first term on the right-hand side is easily computed, the factor Q_Du is then obtained by computing $\widetilde{u}$ from

$$P_DA\widetilde{u} = P_Db \tag{1.6}$$

and then premultiplying it by Q_D. To solve (1.6) we apply a Krylov subspace method for nonsymmetric systems, e.g., GMRES or Bi-CGSTAB [25]. In case of (1.6), however, the system is singular. A singular system can still, however, be solved as long as it is consistent (i.e., $b \in \mathcal{R}(A)$). This is actually the case for (1.6) because the

same projection is applied to both sides. Furthermore, Brown and Walker [3] noted that the least-squares problems in GMRES will give a solution without breakdown if $\mathcal{N}(A) = \mathcal{N}(A^T)$ or if $\mathcal{N}(A) \cap \mathcal{R}(A) = \{0\}$, even though $\mathcal{N}(A) \neq \mathcal{N}(A^T)$.

For symmetric systems the balancing preconditioner was proposed by Mandel [13]. It is used in domain decomposition methods and has been analyzed by several other authors in [14, 7, 22, 21, 24]. For nonsymmetric systems we consider the [illegible] balancing preconditioner of the form

$$P_B = Q_D M^{-1} P_D + Z E^{-1} Y^T, \tag{1.7}$$

with M a nonsingular and possibly nonsymmetric preconditioning matrix. For spd cases (Q_D is replaced by P_D^T in (1.7)), this preconditioner has already been compared with deflation in [17]. With respect to preconditioning with M, the deflated preconditioning system can be written as

$$M^{-1} P_D A u = M^{-1} P_D b. \tag{1.8}$$

As mentioned above, in general some assumptions have to be satisfied to guarantee that GMRES will converge for nonsingular systems. However, we will prove that GMRES applied to (1.6) and (1.8) will converge without any further assumption.

We first compare spectral properties of deflation and the balancing preconditioner. We prove that $P_B A$ and $M^{-1} P_D A$ have the same spectra except for the first r eigenvalues. With this information, bounds of GMRES convergence can be derived. These are presented in section 2. In section 3, GMRES residuals for $M^{-1} P_D A$ and $P_B A$ are compared. With a special starting vector, a relation between residuals of GMRES combined with deflation and the abstract balancing preconditioner can be established for arbitrary full ranked Z and Y. We prove that the preconditioned residual obtained by using the deflation method is less than or equal to the preconditioned residual obtained by using the abstract balancing method. Numerical examples are shown in section 4 for the convection-diffusion equation. Finally, conclusions are drawn in section 5.

2. Spectral properties and GMRES convergence bounds. In this section we evaluate spectral properties of $P_D A$ and their connections with convergence bound of GMRES, and then compare them with $P_B A$. Before doing so, we recall in the following lemma some properties related to P_D and Q_D, whose proofs are easily shown by direct computation.

LEMMA 2.1. [illegible] P_D [illegible] Q_D [illegible] (1.4) [illegible] $Z, Y \in \mathbb{R}^{n \times r}$ [illegible] r [illegible]

(i) $P_D A Z = 0$, $Y^T A Q_D = 0$
(ii) $Q_D Z = 0$, $Y^T P_D = 0$
(iii) $P_D A = A Q_D$

Next we present the convergence bound of GMRES for the unpreconditioned system $Au = b$ due to Saad and Schultz [23], along with its proof.

[illegible] 2.2. $A \in \mathbb{R}^{n \times n}$ is nonsymmetric and diagonalizable, with spectral decomposition $A = X \Lambda X^{-1}$. Here, $X = [x_1 \dots x_n]$ are right eigenvectors of A and $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$, satisfying $A x_i = \lambda_i x_i$, $i = 1, \dots, n$. The eigenvalues λ_i are assumed to be real and nondefective, and $0 < \lambda_i < \lambda_j$ for $i < j$.

THEOREM 2.3. [illegible] A [illegible] 2.2 [illegible] k [illegible] $Au = b$ [illegible] u_0 [illegible]

$$\|r_k\|_2 \leq \kappa_2(X) \epsilon^k \|r_0\|_2, \tag{2.1}$$

... $\kappa_2(X) = \|X\|_2\|X^{-1}\|_2$ ... X, $r_0 = b - Au_0$ ...

$$\epsilon^k = \min_{p\in\mathbb{P}^k, p(0)=1} \max_{i=1,\dots,n} |p(\lambda_i)|, \tag{2.2}$$

... p ... $k-1$ ... $p(0) = 1$ ...

... Let p be a polynomial of degree no larger than $k-1$ with constraint $p(0) = 1$, and let u be a vector in $\mathcal{K}^k$ associated with the residual $b - Au = p(A)r_0$. Then, for $A = X\Lambda X^{-1}$,

$$\|b - Au\|_2 = \|p(A)r_0\|_2 = \|Xp(\Lambda)X^{-1}r_0\|_2 \le \|X\|_2\|X^{-1}\|_2\|\|p(\Lambda)\|_2\|r_0\|_2.$$

Since Λ is a diagonal matrix, $\|p(\Lambda)\|_2 = \max_{i=1,\dots,n} |p(\lambda_i)|$. Consider now u_k, extracted from $\mathcal{K}^k$ but now related to GMRES approximation. Since u_k minimizes the 2-norm of the residual over $u_0 + \mathcal{K}^k$, then for any polynomial p

$$\|b - Au_k\|_2 \le \|b - Au\|_2 \le \|X\|_2\|X^{-1}\|_2\|\|r_0\|_2 \max_{i=1,\dots,n} |p(\lambda_i)|. \tag{2.3}$$

Choosing a polynomial which minimizes the right-hand side, one has

$$\|b - Au_k\|_2 \le \|X\|_2\|X^{-1}\|_2\|\|r_0\|_2 \min_{p\in\mathbb{P}^k, p(0)=1} \max_{i=1,\dots,n} |p(\lambda_i)|,$$

leading to the theorem, with $\|X\|_2\|X^{-1}\|_2 =: \kappa_2(X)$. ∎

A further result is obtained by considering the min-max problem above as the shifted and scaled Chebyshev polynomial in the interval $[\lambda_1, \lambda_n]$,

$$\hat{C}_k(t) = \frac{C_k\left(1 + 2\frac{\lambda_1 - t}{\lambda_n - \lambda_1}\right)}{C_k\left(1 + 2\frac{\lambda_1 - \gamma}{\lambda_n - \lambda_1}\right)}, \tag{2.4}$$

with $\gamma < \lambda_1$. In this case, setting $\gamma = 0$, the constraint results in

$$\min_{p\in\mathbb{P}^k, p(0)=1} \max_{i=1,\dots,n} |p(\lambda_i)| = \min_{p\in\mathbb{P}^k, p(0)=1} \max_{\lambda\in[\lambda_1,\lambda_n]} |p(\lambda_i)| = \frac{1}{C_k\left(2\frac{\mu}{\lambda_n - \lambda_1}\right)}, \tag{2.5}$$

where $\mu = (\lambda_1 + \lambda_n)/2$. As $C_k(t)$ can alternatively be written as

$$C_k(t) = \frac{1}{2}\left[\left(t + \sqrt{t^2-1}\right)^k + \left(t + \sqrt{t^2-1}\right)^{-k}\right] \ge \frac{1}{2}\left(t + \sqrt{t^2-1}\right)^k, \tag{2.6}$$

we see, after some manipulations, that $C_k(2\mu/(\lambda_n - \lambda_1)) \ge 1 - 1/(\sqrt{\kappa} + 1)$, where $\kappa = \lambda_n/\lambda_1$. We then have the following corollary.

COROLLARY 2.4. ... A ... 2.2 ... $\kappa := \lambda_n/\lambda_1$ ... k ...

$$\|r_k\|_2 \le 2\kappa_2(X)\left(1 - \frac{2}{\sqrt{\kappa} + 1}\right)^k \|r_0\|_2. \tag{2.7}$$

This convergence bound is similar to the convergence bound of CG, except that the bound is now represented by the 2-norm of residuals. If A is spd, then $\kappa_2(X) = 1$. For more general cases, $\kappa_2(X)$ is not known and is too expensive to compute. Furthermore, $\kappa = \lambda_n/\lambda_1$ is not the condition number as usually referred to in case of spd systems. Under Assumption 2.2 we may still, however, associate κ in (2.7) with the quality of eigenvalues clustering. The smaller the value is, the more clustered the eigenvalues are.

2.1. Spectrum of deflation and balancing preconditioners. In this section we compare spectral properties of $P_D A$ and $P_B A$. For generality, we consider the deflated linear system (1.8), with M any nonsingular and possibly nonsymmetric matrix. The eigenvalues of A and $M^{-1}A$ are in general not known.

LEMMA 2.5. [illegible] A [illegible] M [illegible] Z [illegible] Y, $M^{-1}P_D A$ [illegible] $P_B A$ [illegible] r [illegible] 0 [illegible] 1 [illegible].

[illegible]. For deflation, for any $Z \in \mathbb{R}^{n\times r}$, $M^{-1}P_D AZ = 0$ because $P_D AZ = 0$. Thus, r eigenvalues are equal to 0. By using the definition of P_B, we have $P_B AZ = Q_D M^{-1}P_D AZ + ZE^{-1}Y^T AZ = Z$, because $Y^T AZ = E$. In this case, r eigenvalues are equal to 1.

In both cases, $Z = [z_1, \ldots, z_r]$ are the eigenvectors of $M^{-1}P_D A$ and $P_B A$ associated with eigenvalues equal to 0 and 1, respectively. ∎

In Krylov subspace methods, a preconditioner is chosen such that the preconditioned linear system is better conditioned than the original system. With an effective preconditioner M, we usually have $\sigma(M^{-1}A) \neq \sigma(A)$, but $\kappa(M^{-1}A) \leq \kappa(A)$. An explicit relation between $\sigma(M^{-1}A)$ and $\sigma(A)$ is generally not known and is often difficult to determine. This is also the case with $M^{-1}P_D A$. This explicit relation is, however, not necessary in this paper as our purpose is mainly to compare deflation and the abstract balancing preconditioner. To result in such a comparison, we need some intermediate results.

By using Lemma 2.5, we denote the spectrum of $M^{-1}P_D A$ by $\sigma(M^{-1}P_D A) = \{0, \ldots, 0, \mu_{r+1}, \ldots, \mu_n\}$, where in general $\mu_i \neq \lambda_i$, $\lambda_i \in \sigma(A)$, $r+1 \leq i \leq n$. Regarding $M^{-1}P_D A$, we have the following spectral equivalence.

LEMMA 2.6. [illegible] A, M [illegible] $Y^T AZ$ [illegible]

$$\sigma(Q_D M^{-1}P_D A) = \sigma(M^{-1}P_D A) = \sigma(Q_D M^{-1}A). \tag{2.8}$$

[illegible]. We have

$$\sigma(Q_D M^{-1}P_D A) = \sigma(M^{-1}P_D AQ_D) = \sigma(M^{-1}P_D^2 A) = \sigma(M^{-1}P_D A),$$

which proves the first equality. The second equality is proved in a similar way. ∎

LEMMA 2.7. [illegible] A, M [illegible] $Y^T AZ$ [illegible]

$$\sigma((Q_D M^{-1}P_D + ZE^{-1}Y^T)A) = \sigma((M^{-1}P_D + ZE^{-1}Y^T)A). \tag{2.9}$$

[illegible]. We note that

$$\begin{aligned}(M^{-1}P_D + ZE^{-1}Y^T)A &= M^{-1}P_D^2 A + ZE^{-1}Y^T A \\ &= M^{-1}P_D AQ_D + ZE^{-1}Y^T A \\ &= (M^{-1}P_D A - I)Q_D + I. \end{aligned} \tag{2.10}$$

Hence

$$\lambda \in \sigma((M^{-1}P_D + ZE^{-1}Y^T)A) \Leftrightarrow \lambda = \mu + 1 \quad \text{for } \mu \in \sigma((M^{-1}P_D A - I)Q_D).$$

However,

$$\sigma((M^{-1}P_D A - I)Q_D) = \sigma(Q_D(M^{-1}P_D A - I)) = \sigma(Q_D M^{-1}P_D A - Q_D).$$

Thus

$$\begin{aligned}\sigma((M^{-1}P_D + ZE^{-1}Y^T)A) &= \sigma(Q_D M^{-1}P_D A - Q_D + I)\\ &= \sigma(Q_D M^{-1}P_D A + ZE^{-1}Y^T A)\\ &= \sigma((Q_D M^{-1}P_D + ZE^{-1}Y^T)A),\end{aligned} \tag{2.11}$$

which proves the lemma. ∎

We then have the following spectral relation between $M^{-1}P_D A$ and $P_B A$.

THEOREM 2.8. [illegible] $M^{-1}P_D A$ [illegible]

$$\sigma(M^{-1}P_D A) = \{0, \ldots, 0, \mu_{r+1}, \ldots, \mu_n\}.$$

[illegible]

$$\sigma(P_B A) = \{1, \ldots, 1, \mu_{r+1}, \ldots, \mu_n\}.$$

[illegible] $P_B A$ [illegible]

$$\sigma(P_B A) = \{1, \ldots, 1, \mu_{r+1}, \ldots, \mu_n\},$$

[illegible]

$$\sigma(M^{-1}P_D A) = \{0, \ldots, 0, \mu_{r+1}, \ldots, \mu_n\}.$$

[illegible]. For $i = 1, \ldots, r$, $P_B AZ = Q_D M^{-1}P_D AZ + ZE^{-1}Y^T AZ = Z$, because $P_D AZ = 0$. Hence, the eigenvectors of $P_B AZ$ which correspond to eigenvalues equal to 1 are the same as those corresponding to eigenvalues equal to 0 of $M^{-1}P_D A$, i.e., $Z = [z_1 \ldots z_r]$.

For $r + 1 \le i \le n$, suppose that $\widetilde{v}_i$ satisfies $M^{-1}P_D A\widetilde{v}_i = \mu_i \widetilde{v}_i$, where μ_i is the corresponding eigenvalue. In this case, we have that $M^{-1}P_D A\widetilde{v}_i = M^{-1}P_D^2 A\widetilde{v}_i = M^{-1}P_D AQ_D\widetilde{v}_i = \mu_i\widetilde{v}_i \neq 0$, implying that $Q_D\widetilde{v}_i \neq 0$ for $\mu_i \neq 0$. Thus,

$$\begin{aligned}P_B AQ_D\widetilde{v}_i &= Q_D M^{-1}P_D AQ_D\widetilde{v}_i + ZE^{-1}Y^T AQ_D\widetilde{v}_i = Q_D M^{-1}P_D^2 A\widetilde{v}_i\\ &= Q_D M^{-1}P_D A\widetilde{v}_i = \mu_i Q_D\widetilde{v}_i.\end{aligned}$$

Hence, for $i = r + 1, \ldots, n$, the eigenvalues of $P_B A$ are the same as the eigenvalues of $M^{-1}P_D A$, with eigenvectors $Q_D\widetilde{v}_i$.

To prove the second statement, we know that for $i = 1, \ldots, r$, $P_B AZ = Z$, which gives, by expanding P_B, $Q_D M^{-1}P_D AZ = 0$. Hence, $0 \in \sigma(Q_D M^{-1}P_D A)$, implying $0 \in \sigma(M^{-1}P_D A)$ due to Lemma 2.6.

For $i = r + 1, \ldots, n$, notice that

$$P_B A\widetilde{v}_i = Q_D M^{-1}P_D A\widetilde{v}_i + ZE^{-1}Y^T A\widetilde{v}_i = \mu_i\widetilde{v}_i$$

implies

$$Q_D P_B A\widetilde{v}_i = Q_D M^{-1}P_D A\widetilde{v}_i = Q_D M^{-1}P_D AQ_D\widetilde{v}_i = \mu_i Q_D\widetilde{v}_i,$$

because $Q_D Z = 0$. Thus, μ_i is an eigenvalue of $Q_D M^{-1}P_D A$. However, due to Lemma 2.6, it is also an eigenvalue of $M^{-1}P_D A$. This completes the proof. ∎

Thus, for any full ranked Z and Y and any nonsingular matrix M, deflation and the balancing preconditioner have similar spectra. Furthermore, $P_B A$ has eigenvectors $Q_D\widetilde{v}_i$, with $\widetilde{v}_i$ the eigenvectors of $M^{-1}P_D A$.

THEOREM 2.9. [illegible] Z [illegible] Y [illegible] M [illegible] P_BA [illegible] $M^{-1}P_DA$ [illegible] P_BA [illegible]. Assume that P_BA is singular. Then there is a vector $x_i \neq 0$ such that $P_BAx_i = \mu_i \cdot x_i = 0$. So, $\mu_i = 0$. From the proof of Theorem 2.8, we know that $x_i = Q_D\widetilde{v}_i$, where $\widetilde{v}_i$ is the eigenvector of $M^{-1}P_DA$ associated with $\mu_i = 0$. Therefore, $M^{-1}P_DA\widetilde{v}_i = 0 \cdot \widetilde{v}_i = 0$. Since M is nonsingular, $P_DA\widetilde{v}_i = 0$, which immediately implies $A\widetilde{v}_i = AZE^{-1}Y^TA\widetilde{v}_i$ or $\widetilde{v}_i = ZE^{-1}Y^TA\widetilde{v}_i$, due to the definition of P_D. In this case, $x_i = Q_D\widetilde{v}_i = Q_DZE^{-1}Y^TA\widetilde{v}_i = 0$, because $Q_DZ = 0$, which does not satisfy the assumption. Hence, $\mu_i = 0$ is not an eigenvalue of P_BA, and henceforth P_BA is nonsingular.

Now suppose that $M^{-1}P_DA\widetilde{v}_i = 0$. Thus, we have

$$P_BA\widetilde{v}_i = Q_DM^{-1}P_DA\widetilde{v}_i + ZE^{-1}Y^TA\widetilde{v}_i = ZE^{-1}Y^TA\widetilde{v}_i. \tag{2.12}$$

This means that $P_BA\widetilde{v}_i = \widetilde{v}_i$ because $\widetilde{v}_i = ZE^{-1}Y^TA\widetilde{v}_i$ as above. Hence, any zero eigenvalues in case of deflation are shifted to one by the balancing preconditioner. ∎

For completeness, we consider a special case where $M = I$, and Z and Y satisfy the following assumption.

[illegible] 2.10. We set $Z = [v_1 \dots v_r]$, where $Av_i = \lambda_iv_i$, $i = 1, \dots, r$. Also, we set $Y = [w_1 \dots w_r]$ the left eigenvector matrix of A, determined from $w_i^TA = \lambda_iw_i^T$ and chosen such that $Y^TZ = I_r$, where I_r is the identity matrix of dimension r. For the left eigenvectors, $W = [w_1 \dots w_n]$.

THEOREM 2.11. [illegible] Z [illegible] Y [illegible] 2.10 [illegible] $M = I$ [illegible] P_B [illegible]

$$\sigma(P_DA) = \{0, \dots, 0, \lambda_{r+1}, \dots, \lambda_n\},$$
$$\sigma(P_BA) = \{1, \dots, 1, \lambda_{r+1}, \dots, \lambda_n\}.$$

[illegible]. Under Assumption 2.10, obviously $E = Y^TAZ = \mathrm{diag}(\lambda_1, \dots, \lambda_r) =: \Lambda_r$. For deflation, we see that for $i = 1, \dots, r$

$$P_DAv_i = \left(I - AZ\Lambda_r^{-1}Y^T\right)Av_i = \lambda_iv_i - Z\Lambda_rY^Tv_i = 0,$$

because $Y^TZ = I$. Similarly, for $i = r+1, \dots, n$, $P_DAv_i = \lambda_iv_i - Z\Lambda_rY^Tv_i = \lambda_iv_i$, because $Y^Tv_i = 0$. This leads to the first result.

For the balancing preconditioner, one can also proceed with the same procedure as above. In this case

$$P_BAv_i = (I - ZE^{-1}Y^TA)(I - AZE^{-1}Y^T)Av_i + ZE^{-1}Y^TAv_i. \tag{2.13}$$

By expanding (2.13) and making use of orthogonality of eigenvectors, we have

$$P_BAv_i = v_i, \quad i = 1, \dots, r. \tag{2.14}$$

Again, due to orthogonality we also have $P_BAv_i = \lambda_iv_i$ for $i = r+1, \dots, n$. ∎

So, in this very particular case P_DA and P_BA have similar spectra with A, but with r eigenvalues shifted to 0 and 1, respectively. The rest of the spectrum of A is untouched. Furthermore, P_DA and P_BA share the same eigenvectors, which are equal to the eigenvectors of A.

2.2. GMRES convergence bounds. Next we provide a GMRES convergence bound for $M^{-1}P_DA$ and P_BA. Here we restrict our discussion to the case satisfying Assumptions 2.2 and 2.10, and $M = I$. We note that, because of Assumptions 2.2 and 2.10,

(i) $X = [Z\, X_{n-r}]$, $X_{n-r} = [x_{r+1} \ \dots \ x_n]$, $W = [Y\, W_{n-r}]$, $W_{n-r} = [w_{r+1} \ \dots \ w_n]$, and

(ii) $E = Y^TAZ = Y^TZ\Lambda_r = \Lambda_r$, where $\Lambda_r = \mathrm{diag}(\lambda_1, \dots, \lambda_r)$.

For deflation we have the following lemma.

LEMMA 2.12. [illegible] $\Lambda_D = \mathrm{diag}\,(0, \dots, 0, \lambda_{r+1}, \dots, \lambda_n)$ [illegible] $\widetilde{r}_{0,D} = P_D(b - A\widetilde{u}_0)$ [illegible] 2.2 [illegible] 2.10 [illegible] k [illegible] $P_DA\widetilde{u} = P_Db$.

$$\mathcal{K}^k(P_DA, \widetilde{r}_{0,D}) = \{\widetilde{r}_{0,D}, X\Lambda_DX^{-1}\widetilde{r}_{0,D}, \dots, X\Lambda_D^{k-1}X^{-1}\widetilde{r}_{0,D}\}. \tag{2.15}$$

[illegible]. For $k = 1$, $\widetilde{r}_0 = P_D(b - A\widetilde{u}_{0,D})$. For $k = 2$,

$$P_DA = (I - AZE^{-1}Y^T)A = A - AZE^{-1}Y^TA = X(I - \Lambda X^{-1}ZE^{-1}Y^TX)\Lambda X^{-1}.$$

Note that, because $W^TX = I$, where $A^TW = W\Lambda$,

$$\begin{aligned} X^{-1}ZE^{-1}Y^TX = W^TZE^{-1}Y^TX &= \begin{bmatrix} Y^T \\ W_{n-r}^T \end{bmatrix} Z\Lambda^{-1}Y^T \begin{bmatrix} Z & X_{n-r} \end{bmatrix}, \\ &= \begin{bmatrix} Y^TZ\Lambda^{-1} \\ W_{n-r}^TZ\Lambda^{-1} \end{bmatrix} \begin{bmatrix} Y^TZ & Y^TX_{n-r} \end{bmatrix} = \begin{bmatrix} \Lambda^{-1} \\ 0 \end{bmatrix} \begin{bmatrix} I_r & 0 \end{bmatrix}, \\ &= \begin{bmatrix} \Lambda_r^{-1} & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned} \tag{2.16}$$

Thus,

$$\begin{aligned} P_DA &= X\left(I - \Lambda\begin{bmatrix} \Lambda_r^{-1} & 0 \\ 0 & 0 \end{bmatrix}\right)\Lambda X^{-1} = X\left(I - \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}\right)\begin{bmatrix} \Lambda_r & 0 \\ 0 & \Lambda_{n-r} \end{bmatrix}X^{-1}, \\ &= X\Lambda_DX^{-1}. \end{aligned}$$

For $k = 3$, $(P_DA)^2 = P_DAP_DA = X\Lambda_DX^{-1}X\Lambda_DX^{-1} = X\Lambda_D^2X^{-1}$. By repeating the computation for $l = 4, \dots, k-1$ the desired result is obtained. ∎

THEOREM 2.13. [illegible] Z [illegible] Y [illegible] 2.2 [illegible] 2.10 [illegible] $M = I$ [illegible] $\kappa_D = \lambda_n/\lambda_{r+1}$ [illegible] $\widetilde{u}_0$ [illegible] $P_DA\widetilde{u} = P_Db$ [illegible] 2 [illegible]

$$\|\widetilde{r}_{k,D}\|_2 \le 2\kappa_2(X)\left(1 - \frac{2}{\sqrt{\kappa_D} + 1}\right)^k \|\widetilde{r}_{0,D}\|_2, \tag{2.17}$$

[illegible] $\widetilde{r}_{k,D} = P_D(b - Au_k)$ [illegible]. Here, we have $\widetilde{r} = p(P_DA)\widetilde{r}_{0,D} = Xp(\Lambda_D)X^{-1}\widetilde{r}_{0,D}$, due to Lemma 2.12, where p is a polynomial of degree no larger than $k-1$, with $p(0) = 1$. Hence,

$$\|\widetilde{r}\|_2 \le \|X\|_2\|X^{-1}\|_2\|p(\Lambda_D)\|_2\|\widetilde{r}_{0,D}\|_2.$$

Noting that $\widetilde{u}_k$ minimizes the 2-norm of residual over $\widetilde{u}_0 + \mathcal{K}^k(P_DA, \widetilde{r}_{0,D})$ and $\|p(\Lambda_D)\|_2 = \max_{i=r+1,\dots,n}|p(\lambda_i)|$, choosing a polynomial which minimizes the right-hand side leads to

$$\|\widetilde{r}_{k,D}\|_2 := \|P_D(b - A\widetilde{u}_{k,D})\|_2 \le \|X\|_2\|X^{-1}\|_2\|\widetilde{r}_0\|_2 \min_{p\in\mathbb{P}^k, p(0)=1} \max_{i=r+1,\dots,n} |p(\lambda_i)|.$$

Taking the shifted and scaled Chebyshev polynomial as the trial polynomial with $\lambda \in [\lambda_{r+1}, \lambda_n]$ and repeating the same procedure as in the previous section one arrives at the desired inequality, with $\kappa_D = \lambda_n/\lambda_{r+1}$. ∎

We see that A and $P_D A$ share the same eigenvectors. Since $\lambda_{r+1} \geq \lambda_1$, $\kappa_D \leq \kappa$, GMRES with deflation preconditioner will asymptotically converge faster than without deflation preconditioner.

2.14. Note that this comparison is not fair because $\widetilde{r}$ is the residual of the preconditioned system, and not of the original system. In practice one usually is more interested in the residual of the original system, which is not the by-product of the left preconditioning GMRES. A more detailed residual analysis in section 3 reveals, however, that in the way the solution is computed, GMRES combined with deflation produces actual residuals which are the same as the preconditioned residuals; see Theorem 3.1.

For the balancing preconditioner, the Krylov subspace associated with it is written as

$$\begin{aligned} &\mathcal{K}^k(P_B A, P_B(b - Au_0)) \\ &= \operatorname{span}\{P_B(b - Au_0), P_B A P_B(b - Au_0), \ldots, (P_B A)^{k-1} P_B(b - Au_0)\}. \end{aligned} \tag{2.18}$$

For cases under Assumptions 2.2 and 2.10 we have the following lemma.

LEMMA 2.15. $\Lambda_B = \operatorname{diag}(1, \ldots, 1, \lambda_{r+1}, \ldots, \lambda_n)$ $\widetilde{r}_{0,B} = P_B(b - Au_0)$ 2.2 2.10 $M = I$

$$\mathcal{K}^k(P_B A, \widetilde{r}_{0,B}) = \{\widetilde{r}_{0,B}, X\Lambda_B X^{-1}\widetilde{r}_{0,B}, \ldots, X(\Lambda_B)^{k-1}X^{-1}\widetilde{r}_{0,N}\}.$$

. For $k = 1$, $\widetilde{r}_{0,B} := P_B(b - Au_0) = P_B r_0$. For $k = 2$,

$$\begin{aligned} P_B A &= (Q_D P_D + ZE^{-1}Y^T)A = (I - ZE^{-1}Y^T A)(I - AZE^{-1}Y^T)A + ZE^{-1}Y^T A \\ &= A - ZE^{-1}Y^T AA - AZE^{-1}Y^T A + ZE^{-1}Y^T AAZE^{-1}Y^T A + ZE^{-1}Y^T A \\ &= X(I - X^{-1}ZE^{-1}Y^T AX - \Lambda X^{-1}ZE^{-1}Y^T X \\ &\quad + X^{-1}ZE^{-1}Y^T AAZE^{-1}Y^T X + X^{-1}ZE^{-1}Y^T X)\Lambda X^{-1}. \end{aligned}$$

From the proof of Theorem 2.12, we then have that

$$\begin{aligned} P_B A &= X\left(I - \begin{bmatrix} \Lambda_r^{-1} & 0 \\ 0 & 0 \end{bmatrix}\Lambda - \Lambda\begin{bmatrix} \Lambda_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} + X^{-1}ZY^T X + \begin{bmatrix} \Lambda_r^{-1} & 0 \\ 0 & 0 \end{bmatrix}\right)\Lambda X^{-1} \\ &= X\begin{bmatrix} \Lambda_r^{-1} & 0 \\ 0 & I_{n-r} \end{bmatrix}\Lambda X^{-1} = X\begin{bmatrix} I_r & 0 \\ 0 & \Lambda_{n-r} \end{bmatrix}X^{-1} =: X\Lambda_B X^{-1}, \end{aligned}$$

because

$$X^{-1}ZY^T X = W^T ZY^T X = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}.$$

We can also compute $(P_B A)^k$ for $k > 2$. This leads to the above lemma. ∎

Making use of Lemma 2.15, we have the GMRES convergence bound for the balancing preconditioner.

THEOREM 2.16. . Z Y P_B 2.2 2.10 $M = I$ $\kappa_B = \max\{1, \lambda_n\}/\min\{1, \lambda_{r+1}\}$ u_0 2 $P_B A u = P_B b$

$$\|\widetilde{r}_{k,B}\|_2 \leq 2\kappa_2(X)\left(1 - \frac{2}{\sqrt{\kappa_B} + 1}\right)^k \|\widetilde{r}_{0,B}\|_2. \tag{2.19}$$

. Note that $\|\widetilde{r}_B\| = Xp(\Lambda_B)X^{-1}\widetilde{r}_{0,B}$. The proof then follows along the same lines as in Theorem 2.13. □

Comparing Theorems 2.13 and 2.16, it is clear that GMRES combined with deflation has a convergence bound which is lower than or equal to GMRES combined with the balancing preconditioner. Therefore, we may expect that GMRES applied to a deflation-preconditioned linear system will converge faster than GMRES with the balancing preconditioner.

3. Comparison of GMRES residuals. To have a more detailed comparison between P_DA and P_BA, in this section we evaluate the approximate solutions built by GMRES and the related residuals. We first recall that when applied to (1.1), a Krylov subspace method generates an approximation solution in the Krylov subspace $\mathcal{K}^k(A, r_0)$, defined in (1.2). In case of left preconditioning, the subspace is now spanned by vectors related to the preconditioned system. For

$$BAu = Bb, \tag{3.1}$$

with B any preconditioner, the Krylov subspace related to the initial residual $\widetilde{r}_0 = B(b - Au_0)$, where u_0 is the starting vector, is given by

$$\mathcal{K}^k(BA, \widetilde{r}_0) = \text{span}\{\widetilde{r}_0, BA\widetilde{r}_0, \ldots, (BA)^{k-1}\widetilde{r}_0\}. \tag{3.2}$$

GMRES then minimizes the residual norm

$$\|B(b - A\eta)\|_2, \tag{3.3}$$

where $\eta \in u_0 + \mathcal{K}^k(BA, \widetilde{r}_0)$. The approximate solution is determined by

$$u_k = u_0 + p_{k-1}(BA)\widetilde{r}_0, \tag{3.4}$$

where p_{k-1} is the polynomial of degree $k-1$, which minimizes the residual norm (3.3) among all other polynomials of degree $\leq k-1$.

In deflation, we know that the approximate solution at the kth iteration is obtained from the relation

$$u_{k,D} = (I - Q_D)u + Q_D\widetilde{u}_{k,D} = ZE^{-1}Y^Tb + Q_D\widetilde{u}_{k,D}, \tag{3.5}$$

where $\widetilde{u}_{k,D}$ is computed iteratively from

$$M^{-1}P_DA\widetilde{u} = M^{-1}P_Db, \tag{3.6}$$

with M an appropriate nonsingular and possibly nonsymmetric preconditioning matrix. If GMRES is used in (3.6), the approximate solution extracted from the affine subspace

$$\widetilde{u}_{0,D} + \mathcal{K}^k(M^{-1}P_DA, \widetilde{r}_{0,D}) \tag{3.7}$$

minimizes the 2-norm of the residual $\widetilde{r}_{k,D} = M^{-1}P_D(b - A\eta)$, where $\eta \in \widetilde{u}_{0,D} + \mathcal{K}^k(M^{-1}P_D, \widetilde{r}_{0,D})$. This approximate solution does not, however, minimize the 2-norm of the actual residual $r := b - Au_{k,D}$. A similar situation is also encountered in the case of the abstract balancing preconditioner.

For deflation, however, the following residual relation holds.

THEOREM 3.1. [illegible] Z, Y [illegible] M [illegible] $\widetilde{u}_{0,D} = 0$ [illegible]

$$\begin{aligned} \|M^{-1}r_{k,D}\|_2 &= \|M^{-1}P_D(b - A\widetilde{u}_{k,D})\|_2 \\ &= \min_{\eta \in \mathcal{K}^k(M^{-1}P_DA, M^{-1}P_Db)} \|M^{-1}P_D(b - A\eta)\|_2. \end{aligned} \tag{3.8}$$

[illegible]. By construction, with $\widetilde{u}_{0,D} = 0$,

$$r_{k,D} =: b - Au_{k,D} = b - A(ZE^{-1}Y^Tb + Q_D\widetilde{u}_{k,D}) = P_D(b - A\widetilde{u}_{k,D}),$$

where $\widetilde{u}_{k,D} \in \mathcal{K}^k(M^{-1}P_DA, M^{-1}P_Db)$. Premultiplying the above equality by M^{-1}, we have

$$\|M^{-1}r_{k,D}\|_2 = \|M^{-1}P_D(b - A\widetilde{u}_{k,D})\|_2 = \min_{\eta \in \mathcal{K}^k(M^{-1}P_DA, M^{-1}P_Db)} \|M^{-1}P_D(b - A\eta)\|_2.$$

Clearly, the optimality property holds only for the preconditioned residuals. ∎

By Theorem 3.1, we immediately see that in the case $M = I$, $\|r_{k,D}\|_2 = \|P_D(b - A\widetilde{u}_{k,D})\|_2$; i.e., the actual residual is equal to the deflated residual.

Next we compare the GMRES residuals for deflation and the abstract balancing preconditioner. For nonsymmetric cases with arbitrary starting vector u_0 such a comparison is, however, difficult. Nevertheless, it is still practically useful to make a comparison for specially chosen starting vectors.

We consider $u_{0,B} = ZE^{-1}Y^Tb$ and $\widetilde{u}_{0,D} = 0$. Such a choice of $u_{0,B}$ has particular reasons in terms of implementation. As one notices from the definition of P_B, with a naive implementation, the balancing preconditioner requires two more matrix-vector multiplications than deflation. If A is symmetric positive definite, this choice of $u_{0,B}$ greatly simplifies the CG algorithm and reduces the amount of work of the balancing preconditioner to only one matrix-vector multiplication, which is the same as deflation; see [13, 24]. As shown in [17], for spd systems, such starting vectors lead to exactly the same A-norm of errors of the CG iterant.

First, we define the Krylov subspace corresponding to P_BA and starting vector $u_{0,B} = ZE^{-1}Y^Tb$.

LEMMA 3.2. [illegible] $u_{0,B} = ZE^{-1}Y^Tb$ [illegible] M [illegible] k [illegible] $ZE^{-1}Y^Tb + \mathcal{K}^k(P_BA, \widetilde{r}_{0,B})$ [illegible]

$$\begin{aligned} &\mathcal{K}^k(P_BA, \widetilde{r}_{0,B}) \\ &= Q_D[illegible]\{M^{-1}P_Db, M^{-1}P_DAM^{-1}P_Db, \ldots, (M^{-1}P_DA)^{k-1}M^{-1}P_Db\}, \end{aligned} \tag{3.9}$$

[illegible] $\widetilde{r}_{0,B} = Q_DM^{-1}P_Db$

[illegible]. For $k = 1$, $\widetilde{r}_{0,B} = P_B(b - Au_{0,B}) = Q_DM^{-1}P_Db$. The proof is done by recursive computations for $(P_BA)^l\widetilde{r}_{0,B}$, $l = 2, \ldots, k$. ∎

THEOREM 3.3. [illegible] Z [illegible] Y [illegible] M [illegible] $u_{0,B} = ZE^{-1}Y^Tb$ [illegible] $\widetilde{u}_{0,D} = 0$ [illegible]

$$\|M^{-1}(b - Au_{k,D})\|_2 \leq \|M^{-1}(b - Au_{k,B})\|_2. \tag{3.10}$$

[illegible]. For the abstract balancing preconditioner, the solution of GMRES at the kth iteration is

$$u_{k,B} \in ZE^{-1}Y^Tb + \mathcal{K}^k(P_BA, \widetilde{r}_{0,B}), \quad \widetilde{r}_{0,B} = P_B(b - Au_{0,B}), \tag{3.11}$$

or, by Lemma 3.2,

$$u_{k,B} = ZE^{-1}Y^T b + Q_D\eta, \qquad \eta \in \mathcal{K}^k(M^{-1}P_DA, M^{-1}P_Db), \tag{3.12}$$

whose residual is $r_{k,B} = b - Au_{k,B} = P_D(b - A\eta)$. Thus, $M^{-1}r_{k,B} = M^{-1}P_D(b - A\eta)$, implying $\|M^{-1}r_{k,B}\|_2 = \|M^{-1}P_D(b - A\eta)\|_2$. If ζ minimizes $\|M^{-1}P_D(b - A\zeta)\|_2$ over the subspace $\mathcal{K}^k(M^{-1}P_DA, M^{-1}P_Db)$, then

$$\|M^{-1}P_D(b - A\eta)\|_2 \geq \min_{\zeta \in \mathcal{K}^k(M^{-1}P_DA, M^{-1}P_Db)} \|M^{-1}P_D(b - A\zeta)\|_2. \tag{3.13}$$

In this case, ζ can be obtained by GMRES applied to $M^{-1}P_DA\widetilde{u} = M^{-1}P_Db$ with zero initial guess. Thus, $\zeta = \widetilde{u}_{k,D}$. By using Theorem 3.1, we finally get

$$\|M^{-1}r_{k,B}\|_2 := \|M^{-1}P_D(b - A\eta)\|_2 \geq \|M^{-1}P_D(b - A\widetilde{u}_{k,D})\|_2 =: \|M^{-1}r_{k,D}\|_2,$$

where $r_{k,D} = b - Au_{k,D}$ and $r_{k,B} = b - Au_{k,B}$. ∎

In Theorem 3.3 we proved that the preconditioned residual obtained by using the deflation method is less than or equal to the preconditioned residual obtained by using the abstract balancing method.

Moreover, Theorem 3.3 guarantees that GMRES preconditioned by the singular deflation preconditioner will also converge without any further assumption.

COROLLARY 3.4. [illegible] Z [illegible] Y [illegible] M [illegible] $\widetilde{u}_{0,D} = 0$ [illegible]

$$M^{-1}P_DA\widetilde{u} = M^{-1}P_Db$$

[illegible]. Since the abstract balancing preconditioner is nonsingular (cf. Theorem 2.9), GMRES applied to

$$P_BAu = P_Bb$$

will converge. By Theorem 3.3 the statement then follows immediately. ∎

4. Numerical examples. In this section we perform numerical experiments to confirm our theoretical results. We base our numerical experiments on the linear systems arising from a finite volume discretization of the steady-state convection-diffusion equation

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} - \nabla \cdot \left(\frac{1}{Pe}\nabla u\right) = f \qquad \text{in } \Omega = (0,1)^2, \tag{4.1}$$

with $Pe > 0$ the Péclet number, and f the forcing term. We first consider the one-dimensional (1D) version of (4.1), in which eigenvalues and eigenvectors of the corresponding linear system can be computed cheaply. Later in this section, a two-dimensional (2D) convection-diffusion problem is also discussed.

4.1. Description of the test problem. For our 1D convection-diffusion problem, we consider an artificial 1D problem with jumps in Pe. Problems with jumps in Pe lead to linear systems with very large difference between the largest and the smallest eigenvalues (in magnitude). In our case, we set

$$Pe(x) = \begin{cases} 1, & 0 \leq x < 0.8, \\ 200, & 0.8 \leq x \leq 1. \end{cases} \tag{4.2}$$

The boundary conditions are $u(0) = 0$ and $u(1) = 1$, which resemble extremely thin boundary layer flows near $x = 1$ and $f = 0$. The 1D convection-diffusion equation is discretized by using the cell-centered finite volume discretization. The convective flux term is approximated by the central discretization. In order to avoid wiggly numerical solutions, the grid is refined at the vicinity of $x = 1$, keeping the mesh Péclet number less than 2 for numerical stability reason. (At this moment we are, however, not concerned with the accuracy of the approximate solutions, and are more interested in the validity of the theoretical results for a specific problem.) In the subdomain where $Pe = 1$, 40 cells are used. In total 200 cells are used. The resultant linear system has real and simple eigenvalues, with 18 of them having value less than 1. The largest eigenvalue is ~ 399.9, while the smallest one is ~ 0.12, giving a ratio of an order of 10^3. The eigenvalues and the associated eigenvectors are computed by the MATLAB routine `eig`.

For the 2D case, we consider the convection-diffusion problem with Dirichlet boundary conditions at $x = 1$ and $y = 0$, and homogeneous Neumann boundary conditions at $x = 0$ and $y = 1$ (see [26]). The equation is discretized by using a finite volume discretization. The convection flux term is approximated by an upwind scheme. The Péclet number is 200. The grid is refined in the y-direction in the vicinity of $y = 0$, while in the x-direction the grid size is kept constant. In this case the eigenvalues of the resultant discretization matrix are easily computed.

4.2. Results. We first consider numerical tests from the 1D convection-diffusion equation. Here, we choose Z based on the subdomain structuring proposed in [18] and used in [10]. Suppose that the domain Ω with index set $\mathcal{I} = \{i | u_i \in \Omega\}$ is partitioned into m nonoverlapping subdomains Ω_j, $j = 1, \ldots, m$, with respective index $\mathcal{I}_j = \{i \in \mathcal{I} | u_i \in \Omega_j\}$. Then Z is defined by

$$z_{ij} = \begin{cases} 1, & i \in \mathcal{I}_j, \\ 0, & i \notin \mathcal{I}_j, \end{cases} \tag{4.3}$$

and $Y = Z$. Particularly in this example we first partition Ω into two subdomains of the same Péclet number. Based on this partition, partitioning is done further until the number of deflation vectors needed is reached.

Figure 4.1 shows the GMRES convergence history with deflation and the abstract balancing preconditioner, and $M = I$. In case of the abstract balancing preconditioner, $u_{0,B} = ZE^{-1}Y^T b$. For deflation, zero initial guess is used. Even though both preconditioners result in almost identical convergence, clearly deflation still produces smaller 2-norms of residuals compared to the abstract balancing preconditioner.

Similar results are obtained for $M \neq I$; see Figure 4.2. In this case, we choose $M = \text{diag}(A)$. GMRES combined with deflation produces a preconditioned residual (i.e., $M^{-1}r_k$) whose 2-norm is smaller than with the abstract balancing preconditioner. For the actual residual, this conclusion does not necessarily hold (see right figure). In the right figure, we observe at some steps that the abstract balancing preconditioner produces a smaller 2-norm of the actual residuals than deflation.

Next we consider the 2D model problem. The domain is partitioned into 10×10 subdomains. Deflation vectors are constructed based on (4.3). We choose $M = \text{diag}(A)$. Figure 4.3 shows convergence results for starting vector $\widetilde{u}_{0,D} = 0$ and $u_{0,B} = ZE^{-1}Y^T b$. In this case, residuals related to deflation and the abstract balancing preconditioners are very similar. Figure 4.4 (left) shows that the preconditioned residual (based on M) of the balancing preconditioner is never smaller than that of

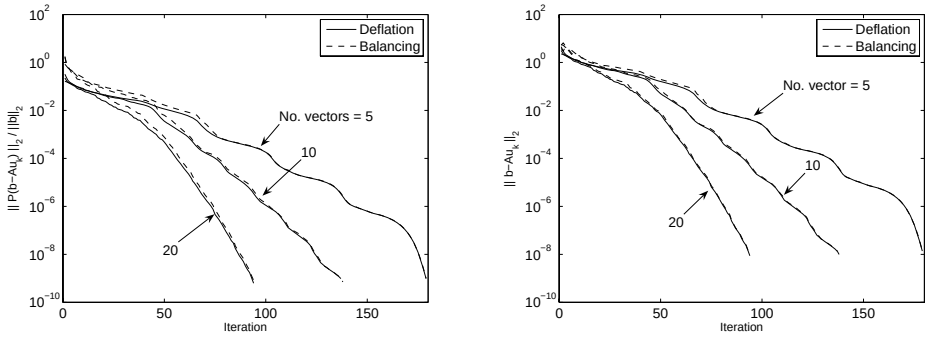


FIG. 4.1. *One-dimensional convection-diffusion equation with jumps. Shown are residuals of the preconditioned system with zero starting vector for deflation and* $u_{0,B} = ZE^{-1}Y^Tb$ *for the balancing preconditioner,* $M = I$*, and* Z *and* Y *consisting of eigenvectors of* A*. Left: preconditioned relative residuals. Right: actual residuals.*

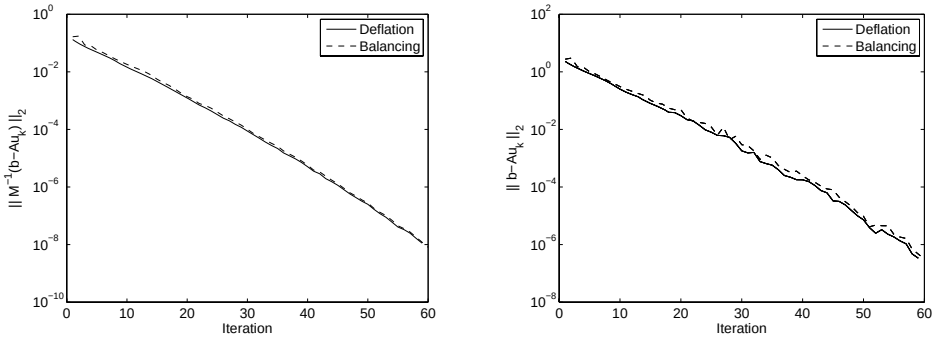


FIG. 4.2. *One-dimensional convection-diffusion equation with jumps. Shown are residuals of the preconditioned system with* $\widetilde{u}_{0,D} = 0$ *and* $u_{0,B} = ZE^{-1}Y^Tb$*,* M *the diagonal scaling preconditioner, and* Z *and* Y *as in* (4.3)*. Left: preconditioned residuals (based on* M*). Right: actual residuals.*

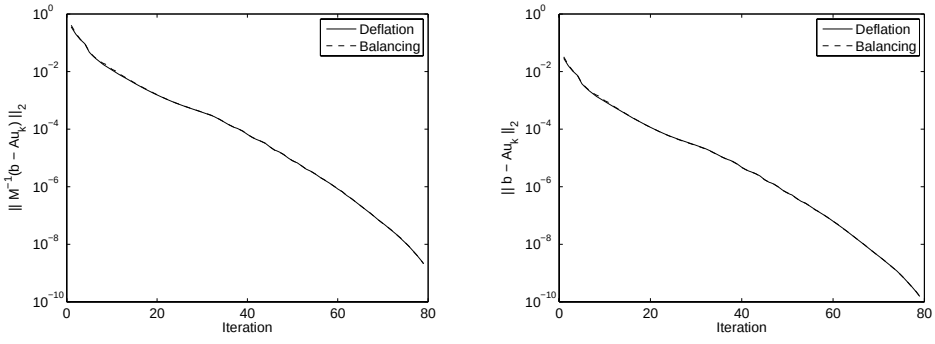


FIG. 4.3. *Two-dimensional convection-diffusion equation with constant coefficient* ($Pe = 200$)*. Shown are residuals of the preconditioned system with* $\widetilde{u}_{0,D} = 0$ *and* $u_{0,B} = ZE^{-1}Y^Tb$*,* M *the diagonal scaling preconditioner, and* Z *and* Y *as in* (4.3)*. Left: preconditioned residuals (based on* M*). Right: actual residuals.*

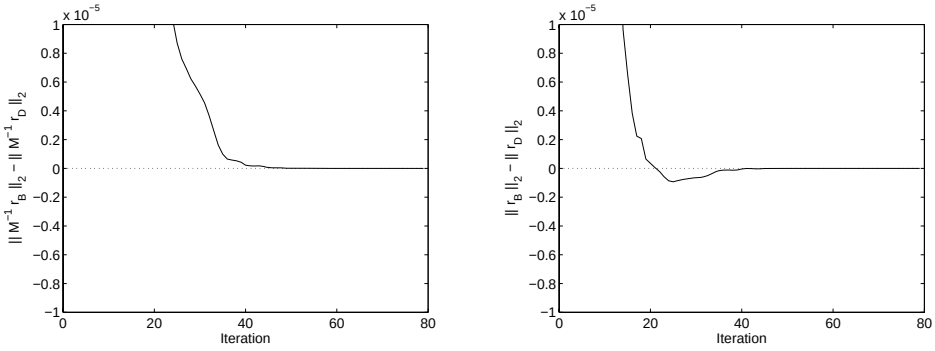


FIG. 4.4. *Residual difference from the* $2D$ *convection-diffusion equation with constant coefficient* ($Pe = 200$). *The starting vectors are* $\widetilde{u}_{0,D} = 0$ *and* $u_{0,B} = ZE^{-1}Y^Tb$, M *the diagonal scaling preconditioner, and* Z *and* Y *as in* (4.3). *Left: preconditioned residuals (based on* M*). Right: actual residuals.*

deflation. This is, however, not always the case for the actual residual (Figure 4.4 (right)).

5. Conclusion. In this paper a comparison between deflation and the abstract balancing preconditioner for nonsymmetric linear systems has been given, within the context of GMRES. We were able to show that many results for symmetric positive definite matrices carry over to some classes of nonsymmetric matrices. We first established that the spectra of the preconditioned systems are similar. Moreover, our analysis shows that with special starting vectors, for nonsingular, nonsymmetric matrix A deflation generates approximate solutions whose related residuals (with respect to M) are never larger than the balancing preconditioner. If the deflation vectors are chosen to be the eigenvectors, the 2-norm of the actual residuals of deflation is always never larger than the balancing preconditioner. For general deflation vectors we proved that the 2-norm of preconditioned residual of GMRES combined with deflation is smaller than that of GMRES combined with the balancing preconditioner. Numerical experiments confirmed the theoretical results and showed that the methods behave similarly.

REFERENCES

[1] B. Aksoylu, H. Klie, and M. Wheeler, *Physics-Based Preconditioners for Porous Media Flow Applications*, ICES Technical Report, The University of Texas, Austin, 2007.

[2] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[3] P. N. Brown and H. F. Walker, *GMRES on (nearly) singular systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 37–51.

[4] A. Chapman and Y. Saad, *Deflated and augmented Krylov subspace techniques*, Numer. Linear Algebra Appl., 4 (1997), pp. 43–66.

[5] M. Clemens, M. Wilke, and T. Weiland, *Extrapolation strategies in numerical schemes for transient magnetic field simulations*, IEEE Trans. Magnetics, 39 (2003), pp. 1171–1174.

[6] M. Clemens, M. Wilke, R. Schuhmann, and T. Weiland, *Subspace projection extrapolation scheme for transient field simulations*, IEEE Trans. Magnetics, 40 (2004), pp. 934–937.

[7] M. Dryja and O. B. Widlund, *Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems*, Comm. Pure Appl. Math., 48 (1995), pp. 121–155.

[8] M. Eiermann, O. G. Ernst, and O. Schneider, *Analysis of acceleration strategies for restarted minimal residual methods*, J. Comput. Appl. Math., 123 (2000), pp. 261–292.

[9] J. Erhel, K. Burrage, and B. Pohl, *Restarted GMRES preconditioned by deflation*, J. Comput. Appl. Math., 69 (1996), pp. 261–292.
[10] J. Frank and C. Vuik, *On the construction of deflation-based preconditioners*, SIAM J. Sci. Comput., 23 (2001), pp. 442–462.
[11] S. A. Kharchenko and A. Y. Yerenim, *Eigenvalue translation based preconditioners for the GMRES(k) method*, Numer. Linear Algebra Appl., 2 (1995), pp. 51–77.
[12] M. E. Kilmer and E. de Sturler, *Recycling subspace information for diffuse optical tomography*, SIAM J. Sci. Comput., 27 (2006), pp. 2140–2166.
[13] J. Mandel, *Balancing domain decomposition*, Comm. Numer. Methods Engrg., 9 (1993), pp. 233–241.
[14] J. Mandel and M. Brezina, *Balancing domain decomposition for problems with large jumps in coefficients*, Math. Comp., 216 (1996), pp. 1387–1401.
[15] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.
[16] R. Nabben and C. Vuik, *A comparison of deflation and coarse grid correction applied to porous media flow*, SIAM J. Numer. Anal., 42 (2004), pp. 1631–1647.
[17] R. Nabben and C. Vuik, *A comparison of deflation and the balancing preconditioner*, SIAM J. Sci. Comput., 27 (2006), pp. 1742–1759.
[18] R. A. Nicolaides, *Deflation of conjugate gradients with applications to boundary value problems*, SIAM J. Numer. Anal., 24 (1987), pp. 355–365.
[19] A. Padiy, O. Axelsson, and B. Polman, *Generalized augmented matrix preconditioning approach and its application to iterative solution of ill-conditioned algebraic systems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 793–818.
[20] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674.
[21] L. F. Pavarino and O. B. Widlund, *Balancing Neumann-Neumann methods for incompressible stokes equations*, Comm. Pure Appl. Math., 55 (2002), pp. 302–335.
[22] A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, Oxford, UK, 1999.
[23] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
[24] A. Tosseli and O. Widlund, *Domain Decomposition Methods*, Springer-Verlag, Berlin, 2005.
[25] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 631–644.
[26] P. Wesseling, *Principles of Computational Fluid Dynamics*, Springer-Verlag, Berlin, Heidelberg, 2001.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 700–712

REACHABILITY AND HOLDABILITY OF NONNEGATIVE STATES*

DIMITRIOS NOUTSOS† AND MICHAEL J. TSATSOMEROS‡

Abstract. Linear differential systems $\dot{x}(t) = Ax(t)$ ($A \in \mathbb{R}^{n\times n}$, $x_0 = x(0) \in \mathbb{R}^n$, $t \geq 0$) whose solutions become and remain nonnegative are studied. It is shown that the eigenvalue of A furthest to the right must be real and must possess nonnegative right and left eigenvectors. Moreover, for some $a \geq 0$, $A + aI$ must be eventually nonnegative, that is, its powers must become and remain entrywise nonnegative. Initial conditions x_0 that result in nonnegative states $x(t)$ in finite time are shown to form a convex cone that is related to the matrix exponential e^{tA} and its eventual nonnegativity.

Key words. eventually nonnegative matrix, exponentially nonnegative matrix, point of nonnegative potential, Perron–Frobenius, Metzler matrix, convex cone

AMS subject classifications. 15A48, 93B03, 65F10

DOI. 10.1137/070693850

1. Introduction.

In dynamical systems theory, one is frequently interested in qualitative information regarding state evolution. In particular, due to physical and modeling constraints arising in engineering, biological, medical, behavioral, and economic applications, it is commonly of interest to impose or consider conditions for nonnegativity of the states; see, e.g., [2, 6]. Such applications typically draw on the theory, or directly take the form, of a linear differential system,

$$\dot{x}(t) = Ax(t), \quad A \in \mathbb{R}^{n\times n}, \quad x(0) = x_0 \in \mathbb{R}^n, \quad t \geq 0, \tag{1.1}$$

whose solution is given by $x(t) = e^{tA}x_0$. We shall refer here to the set

$$\{x(t) = e^{tA}x_0 \mid t \in [0,\infty)\}$$

as the [illegible] x_0 and say that x_0 gives rise to this trajectory. In this paper we will consider conditions for the entrywise nonnegativity of the trajectories associated with (1.1). Our main concern is the following "hit and hold" problem:

[illegible] x_0 [illegible]

More specifically, we will seek characterizations of system parameters that lead to a trajectory becoming nonnegative at a finite time ([illegible] $\mathbb{R}^n_+$) and remaining nonnegative for all time thereafter ([illegible] $\mathbb{R}^n_+$). This endeavor will comprise two related efforts:

(1) Study matrices $A \in \mathbb{R}^{n\times n}$ for which there exists $t_0 \in [0,\infty)$ such that $e^{tA} \geq 0$ for all $t \geq t_0$. We shall term such matrices [illegible].

(2) Given an eventually exponentially nonnegative matrix A, study initial points $x_0 \in \mathbb{R}^n$ for which there exists $\hat{t} \in [0,\infty)$ such that $e^{tA}x_0 \geq 0$ for all $t \geq \hat{t}$. We shall refer to such initial points as [illegible].

Some comments regarding these two goals and the structure of this paper are in order.

*Received by the editors June 6, 2007; accepted for publication (in revised form) by D. Szyld February 22, 2008; published electronically June 18, 2008.
http://www.siam.org/journals/simax/30-2/69385.html

†Department of Mathematics, University of Ioannina, GR–451 10, Ioannina, Greece (dnoutsos@uoi.gr).

‡Department of Mathematics, Washington State University, Pullman, WA 99164 (tsat@wsu.edu).

First, matrices all of whose off-diagonal entries are nonnegative (known as essentially nonnegative or Metzler matrices) are eventually exponentially nonnegative (with $t_0 = 0$). However, as we shall see in section 3, the eventually exponentially nonnegative matrices form a larger matrix class. They are closely related to the eventually nonnegative matrices, namely, matrices whose powers become and remain nonnegative. It is this latter fact that provides further motivation for our study, as eventually nonnegative matrices arise in the theory of positive control systems; see e.g., [16].

Second, it is clear that $\mathbb{R}^n_+$ (the nonnegative orthant) comprises points of nonnegative potential but as we shall see, in the general case, the totality of such points forms a convex cone that strictly contains $\mathbb{R}^n_+$. Our relevant analysis is in section 4, where points of nonnegative potential and the asymptotic behavior of solutions are connected to the matrix exponential e^{tA} and its eventual nonnegativity. We note that even in applications where initial points and states are de facto nonnegative, points of nonnegative potential outside $\mathbb{R}^n_+$ can be of practical interest. For example, suppose that for some $x_0 \in \mathbb{R}^n$, $\hat{x}_0 = Ax_0$ is a point of nonnegative potential. Then there exists $\hat{t} \geq 0$ such that for all $t \geq \hat{t}$,

$$\dot{x}(t) \;=\; \frac{d}{dt}(e^{tA}x_0) \;=\; Ae^{tA}x_0 \;=\; e^{tA}Ax_0 \;=\; e^{tA}\hat{x}_0 \geq 0;$$

that is, the trajectory emanating from x_0 becomes (at $t = \hat{t}$) and remains entrywise nondecreasing. This situation occurs, e.g., when (1.1) models species that reach a symbiotic state after which none of the populations decreases; see [9].

2. Notation, definitions, and preliminaries. Given an $n \times n$ matrix A, the spectrum of A is denoted by $\sigma(A)$ and its spectral radius by $\rho(A) = \max\{|\lambda| \mid \lambda \in \sigma(A)\}$. An eigenvalue λ of A is said to be [illegible] if $|\lambda| = \rho(A)$. The [illegible] of A is defined and denoted by $\lambda(A) := \max\{\mathrm{Re}\,\lambda \mid \lambda \in \sigma(A)\}$. By $\mathrm{index}_0(A)$ we denote the degree of 0 as a root of the minimal polynomial of A. Consequently, when we say $\mathrm{index}_0(A) \leq 1$, we mean that either A is invertible or that the size of the largest nilpotent Jordan block in the Jordan canonical form of A is 1×1.

The [illegible] in $\mathbb{R}^n$, that is, the set of all nonnegative vectors in $\mathbb{R}^n$, is denoted by $\mathbb{R}^n_+$. For $x \in \mathbb{R}^n$, we use the notation $x \geq 0$ interchangeably with $x \in \mathbb{R}^n_+$.

An $n \times n$ matrix A is called [illegible] if there exists a permutation matrix P such that

$$PAP^T \;=\; \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix},$$

where A_{11} and A_{22} are square, nonvacuous matrices. Otherwise, A is called [illegible]. Recall that irreducibility of A is equivalent to the directed graph of A, $G(A)$, being strongly connected, namely, the existence of a path of edges leading from any vertex i to any other vertex j. For details and further terminology regarding directed graphs, see [1].

Every reducible matrix A can be symmetrically permuted to its [illegible]; namely, for every reducible matrix $A \in \mathbb{R}^{n\times n}$, there exists a permutation matrix P such that

$$PAP^T \;=\; \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ 0 & A_{22} & \cdots & A_{2p} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & A_{pp} \end{bmatrix},$$

where each diagonal block A_{jj} $(j = 1, 2, \ldots, p)$ is square and either irreducible or the 1×1 zero matrix. Note that the diagonal blocks in the Frobenius normal form correspond to a partition of the vertices of $G(A)$ into classes of strongly connected vertex subsets (a singleton is considered strongly connected).

To the Frobenius normal form of A above we associate the [illegible] $R(A)$ of A defined as follows: $R(A)$ has p vertices, each one of them corresponding to a strongly connected set of vertices in the directed graph of A. $R(A)$ has a directed edge from i to j if and only if $A_{ij} \neq 0$. In section 3 we will consider the [illegible] of $R(A)$, $\overline{R(A)}$, which is the directed graph obtained from $R(A)$ having an edge from i to j if and only if there is a path from i to j in $R(A)$.

DEFINITION 2.1. [illegible] $n \times n$ [illegible] $A = [a_{ij}]$ [illegible]

- nonnegative (positive) [illegible] $A \geq 0$ $(A > 0)$, $a_{ij} \geq 0$ (> 0) [illegible] i [illegible] j
- essentially nonnegative (positive) [illegible] $A \overset{\text{s}}{\geq} 0$ $(A \overset{\text{s}}{>} 0)$, $a_{ij} \geq 0$ $(a_{ij} > 0)$ [illegible] $i \neq j$
- eventually nonnegative (positive) [illegible] $A \overset{\text{v}}{\geq} 0$ $(A \overset{\text{v}}{>} 0)$, [illegible] [illegible] k_0 [illegible] $A^k \geq 0$ $(A^k > 0)$ [illegible] $k \geq k_0$ [illegible] [illegible] $k_0 = k_0(A)$ [illegible] power index [illegible] A
- exponentially nonnegative (positive) [illegible] $t \geq 0$, $e^{tA} = \sum_{k=0}^{\infty} \frac{t^k A^k}{k!} \geq 0$ $(e^{tA} > 0)$
- eventually exponentially nonnegative (positive) [illegible] $t_0 \in [0, \infty)$ [illegible] [illegible] $t \geq t_0$, $e^{tA} \geq 0$ $(e^{tA} > 0)$ [illegible] [illegible] $t_0 = t_0(A)$ [illegible] exponential index [illegible] A

LEMMA 2.2. [illegible] $A \in \mathbb{R}^{n \times n}$ [illegible]

(i) A [illegible]

(ii) [illegible] $a \in \mathbb{R}$ [illegible] $A + aI$ [illegible]

(iii) [illegible] $a \in \mathbb{R}$, $A + aI$ [illegible]

[illegible] The equivalences follow readily from the fact that as aI and A commute,

$$e^{t(A+aI)} = e^{atI} e^{tA} = e^{at} e^{tA}. \qquad \square$$

We conclude this section with some notions crucial to the analysis in section 3.

DEFINITION 2.3. [illegible] $A \in \mathbb{R}^{n \times n}$ [illegible]

- [illegible] Perron–Frobenius property [illegible] $\rho(A) > 0$, $\rho(A) \in \sigma(A)$ [illegible] [illegible] $\rho(A)$
- [illegible] strong Perron–Frobenius property [illegible] [illegible] [illegible] $\rho(A)$ [illegible]

$$\rho(A) > |\lambda| \quad \text{for all} \quad \lambda \in \sigma(A), \ \lambda \neq \rho(A),$$

[illegible] $\rho(A)$

By the Perron–Frobenius theorem, every nonnilpotent $A \geq 0$ has the Perron–Frobenius property and every primitive $A \geq 0$ has the strong Perron–Frobenius property; see [1].

3. Eventually exponentially nonnegative matrices. There is a well-known equivalence between the notions of exponential nonnegativity and essential nonnegativity; see [1, Chapter 6, Theorem (3.12)]. We include a proof of this result next for completeness.

LEMMA 3.1. $A \in \mathbb{R}^{n\times n}$ [illegible] $A \overset{s}{\geq} 0$ [illegible]. If $A \overset{s}{\geq} 0$, then there exists large enough $\alpha \geq 0$ such that $A + \alpha I \geq 0$. Hence, as A and αI commute, we have that for all $t \geq 0$,

$$e^{tA} = e^{-t\alpha I} e^{t(A+\alpha I)} = e^{-t\alpha} e^{t(A+\alpha I)} \geq 0.$$

Conversely, let $e^{tA} \geq 0$ for all $t \geq 0$ and by way of contradiction suppose that $a_{ij} < 0$ for some $i \neq j$. Then, denoting the entries of A^k by $a_{ij}^{(k)}$, we have

$$(e^{tA})_{ij} = ta_{ij} + \frac{t^2}{2!}a_{ij}^{(2)} + \frac{t^3}{3!}a_{ij}^{(3)} + \cdots.$$

Thus, letting $t \to 0^+$ we have that for some $t > 0$, $(e^{tA})_{ij} < 0$, a contradiction. ∎

As a consequence of the above lemma, every essentially nonnegative matrix A is eventually exponentially nonnegative with exponential index $t_0 = 0$. We proceed with a characterization of eventually exponentially positive matrices based on some recent results proven in [11].

THEOREM 3.2 (see [11, Theorem 2.2]). [illegible] $A \in \mathbb{R}^{n\times n}$ [illegible]

(i) [illegible] A [illegible] A^T [illegible]

(ii) A [illegible]

(iii) A^T [illegible]

Our main result thus far is the following extension of Theorem 3.2.

THEOREM 3.3. [illegible] $A \in \mathbb{R}^{n\times n}$ [illegible]

(i) [illegible] $a \geq 0$ [illegible] $A+aI$ [illegible] A^T+aI [illegible]

(ii) $A + aI$ [illegible] $a \geq 0$

(iii) $A^T + aI$ [illegible] $a \geq 0$

(iv) A [illegible]

(v) A^T [illegible]

. The equivalence of (i)–(iii) is the content of Theorem 3.2 applied to $A+aI$. We will argue the equivalence of (ii) and (iv), with the equivalence of (iii) and (v) being analogous:

Let $A + aI$ be eventually positive and let k_0 be a positive integer such that $(A + aI)^k > 0$ for all $k \geq k_0$. Then there exists large enough $t_0 > 0$ so that the first $k_0 - 1$ terms of the series

$$e^{t(A+aI)} = \sum_{m=0}^{\infty} \frac{t^m (A + aI)^m}{m!}$$

are dominated by the remaining terms, rendering every entry of $e^{t(A+aI)}$ positive for all $t \geq t_0$. It follows that $e^{tA} = e^{-ta}e^{t(A+aI)}$ is positive for all $t \geq t_0$. That is, A is eventually exponentially positive. Conversely, suppose A is eventually exponentially positive. As $(e^A)^k = e^{kA}$, it follows that e^A is eventually positive. Thus, by Theorem 3.2, e^A has the strong Perron–Frobenius property. Recall that $\sigma(e^A) = \{e^\lambda : \lambda \in \sigma(A)\}$ and so $\rho(e^A) = e^\lambda$ for some $\lambda \in \sigma(A)$. Then for each $\mu \in \sigma(A)$ with $\mu \neq \lambda$ we have

$$e^\lambda > |e^\mu| = |e^{\operatorname{Re}\mu + i\operatorname{Im}\mu}| = e^{\operatorname{Re}\mu}.$$

Hence λ is the spectral abscissa of A, namely, $\lambda > \operatorname{Re}\mu$ for all $\mu \in \sigma(A)$ with $\mu \neq \lambda$. In turn, this means that there exists large enough $a > 0$ such that

$$\lambda + a > |\mu + a| \quad \text{for all } \mu \in \sigma(A), \ \ \mu \neq \lambda.$$

As $A + aI$ shares its eigenspaces with e^A, it follows that $A + aI$ has the strong Perron–Frobenius property. Invoking Theorem 3.2 once more, we have that $A + aI$ is eventually positive. □

3.4. Note that the equivalence of (ii) and (iv) in Theorem 3.3 represents a generalization of the fact that $A \overset{s}{>} 0$ is equivalent to A being exponentially positive.

3.5. Consider the matrix

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

and observe that

$$A^2 = \begin{bmatrix} 2 & 3 & 4 & 4 \\ 2 & 3 & 4 & 4 \\ 0 & 1 & 2 & 2 \\ 1 & 2 & 3 & 3 \end{bmatrix}, \quad A^3 = \begin{bmatrix} 5 & 9 & 13 & 13 \\ 5 & 9 & 13 & 13 \\ 1 & 3 & 5 & 5 \\ 3 & 6 & 9 & 9 \end{bmatrix}.$$

It is easily checked that A is an eventually positive matrix with power index $k_0 = 3$, so by Theorem 3.3, A is an eventually exponentially positive matrix. Computing e^{tA} for $t = 1, 2$ we obtain, respectively,

$$\begin{bmatrix} 5.0401 & 6.3618 & 8.6836 & 8.6836 \\ 4.0401 & 7.3618 & 8.6836 & 8.6836 \\ -0.4655 & 2.7873 & 5.0401 & 4.0401 \\ 2.7873 & 3.5746 & 6.3618 & 7.3618 \end{bmatrix}, \begin{bmatrix} 71.2660 & 134.1429 & 198.0199 & 198.0199 \\ 70.2660 & 135.1429 & 198.0199 & 198.0199 \\ 18.4960 & 45.3810 & 71.2660 & 70.2660 \\ 45.3810 & 88.7620 & 134.1429 & 135.1429 \end{bmatrix}.$$

Taking into consideration the location of the nonpositive entries of A and A^2, we infer that the exponential index of A is $t_0 \in (1, 2)$.

Next we focus on eventually exponentially [illegible] matrices and connect them to eventually [illegible] matrices. In what follows we state and prove conditions that are sufficient for eventual exponential nonnegativity and investigate necessary conditions. To do so, we first need to discuss the relationship among the Frobenius normal forms of the powers of an eventually nonnegative matrix. This topic and its relation to the spectrum are studied extensively in [3, 4]. Below we summarize and paraphrase some of these results.

THEOREM 3.6 (see [3, Theorems 3.4 and 3.5]). [illegible] $A \in \mathbb{R}^{n\times n}$ [illegible] $\operatorname{index}_0(A) \leq 1$ [illegible] q [illegible] P [illegible]

(i) $A^k \geq 0$ [illegible] $k \geq q$

(ii) PAP^T [illegible] PA^qP^T [illegible]

(iii) $\overline{R(A)} = \overline{R(A^q)}$

THEOREM 3.7. [illegible] $A \in \mathbb{R}^{n\times n}$ [illegible] $\operatorname{index}_0(A) \leq 1$ [illegible] A [illegible]

[illegible]. To avoid trivialities, suppose $n \geq 2$ and recall Theorem 3.6(ii). Without loss of generality, assume $P = I$; otherwise our considerations apply to a permutational similarity of A. Thus A and A^q are assumed to be in Frobenius normal form

as follows:

$$(3.1)\quad A = \begin{bmatrix} A_{11} & \cdots & \cdots & A_{1p} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A_{pp} \end{bmatrix} \quad \text{and} \quad A^q = \begin{bmatrix} A_{11}^{(q)} & \cdots & \cdots & A_{1p}^{(q)} \\ 0 & \ddots & \ddots & A_{2k}^{(q)} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & A_{pp}^{(q)} \end{bmatrix}.$$

Consider the power series $e^{tA} = \sum_{k=0}^{\infty} \frac{t^m A^k}{k!}$ partitioned in blocks conformably to the matrices in (3.1) and let $c_{ij}(t)$ be the (i,j)th entry of e^{tA}. Abusing slightly the notation, let $\{1, 2, \ldots, p\}$ denote the p strongly connected classes in $G(A)$ implied by (3.1). Let i belong to class u and j to class v, where $u, v \in \{1, 2, \ldots, p\}$. The following cases ensue:

Suppose that $p = 1$. As $n \geq 2$, A is irreducible. Thus for all powers $k \geq q$, the (i,j)th entry of A^k is nonnegative and is indeed positive for at least some powers $\geq q$. As a consequence, as $t \geq 0$ increases, $c_{ij}(t)$ is dominated in the power series by positive terms. That is, $c_{ij}(t)$ becomes and remains positive for all large enough $t \geq 0$.

Suppose next that $p > 1$. The blocks in the lower triangular part of the block partition of each A^k implied by (3.1) must be zero; namely, if $u > v$, then $c_{ij}(t) = 0$ for all $t \geq 0$.

If $u = v$, that is, if i, j belong to the same equivalence class, then either A_{uu} and $A_{uu}^{(q)}$ are both equal to the 1×1 zero matrix or they are both irreducible. In the former case, $c_{ij}(t) = 0$ for all $t \geq 0$, and in the latter case, $c_{ij}(t)$ becomes and remains positive for all large enough $t \geq 0$ analogously to the $p = 1$ case above.

Finally, let us consider the sign of $c_{ij}(t)$ when $u < v$. Let $a_{ij}^{(k)}$ denote the (i,j)th entry of A^k. If $a_{ij}^{(k)} = 0$ for all $k < q$, then by Theorem 3.6(i) we have that $c_{ij}(t) \geq 0$ for all $t \geq 0$. If $a_{ij}^{(k)} \neq 0$ for some $k < q$, then there must be a path form i to j in $G(A)$. Thus there is a path from u to v in $R(A)$. By Theorem 3.6(iii), $\overline{R(A)} = \overline{R(A^q)}$ and so there must be a path from u to v in $R(A^q)$. It follows that there is a path from i to j in $G(A^q)$. In turn, this implies that there is a power $m \geq q$ such that $a_{ij}^{(m)} > 0$. As $a_{ij}^{(k)} \geq 0$ for all $k \geq q$, we have once again that $c_{ij}(t)$ is dominated in the power series by positive terms and so it becomes and remains positive for all large enough $t \geq 0$.

To conclude, we have shown that each entry $c_{ij}(t)$ of e^{tA} becomes and remains nonnegative for all large enough $t \geq 0$, namely, that A is eventually exponentially positive. □

COROLLARY 3.8. [illegible] $A \in \mathbb{R}^{n \times n}$ [illegible] $A + aI$ [illegible] $a \in [a_1, a_2]$ $(a_1 < a_2)$ [illegible] A [illegible].

[illegible]. Since $\sigma(A)$ is a finite set, there exists $a \in [a_1, a_2]$ such that $A + aI$ is invertible. Hence $\text{index}_0(A + aI) = 0$ and so by Theorem 3.7, $A + aI$ is eventually exponentially nonnegative. By Lemma 2.2, it follows that A is eventually exponentially nonnegative. □

We illustrate the above results on eventual nonnegativity with the following examples.

EXAMPLE 3.9. Consider

$$A = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

for which

$$A^2 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 2 \end{bmatrix}, \quad A^3 = \begin{bmatrix} 0 & 1 & 3 & 1 \\ 1 & 0 & 5 & 5 \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 4 & 4 \end{bmatrix},$$

$$A^4 = \begin{bmatrix} 1 & 0 & 5 & 5 \\ 0 & 1 & 11 & 9 \\ 0 & 0 & 8 & 8 \\ 0 & 0 & 8 & 8 \end{bmatrix}, \quad A^5 = \begin{bmatrix} 0 & 1 & 11 & 9 \\ 1 & 0 & 21 & 21 \\ 0 & 0 & 16 & 16 \\ 0 & 0 & 16 & 16 \end{bmatrix}.$$

Notice that A is reducible, eventually nonnegative, and, referring to Theorem 3.6, $q = k_0 = 2$. Since $\text{index}_0(A) = 1$, Theorem 3.7 implies that A is an eventually exponentially nonnegative matrix. For illustration, we compute e^{tA} for $t = 1, 2$ to be, respectively,

$$\begin{bmatrix} 1.5431 & 1.1752 & 2.3404 & -0.0100 \\ 1.1752 & 1.5431 & 4.0487 & 2.9625 \\ 0 & 0 & 4.1945 & 3.1945 \\ 0 & 0 & 3.1945 & 4.1945 \end{bmatrix}, \begin{bmatrix} 3.7622 & 3.6269 & 18.1543 & 10.9006 \\ 3.6269 & 3.7622 & 35.4439 & 29.9195 \\ 0 & 0 & 27.7991 & 26.7991 \\ 0 & 0 & 26.7991 & 27.7991 \end{bmatrix}.$$

This confirms A is an eventually exponentially nonnegative matrix with $1 < t_0 < 2$.

EXAMPLE 3.10. Consider the matrix

$$A = \begin{bmatrix} 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

and its sequence of powers

$$A^k = \begin{bmatrix} 2^{k-1} & 2^{k-1} & 0 & 0 \\ 2^{k-1} & 2^{k-1} & 0 & 0 \\ 0 & 0 & 2^{k-1} & 2^{k-1} \\ 0 & 0 & 2^{k-1} & 2^{k-1} \end{bmatrix} \quad (k = 2, 3, \ldots).$$

The matrix A is eventually nonnegative with $k_0 = 2$. As the $(1, 2)$ block of A^k is 0 for all $k \geq 2$, while the one of A is not and contains negative entries, A is not eventually exponentially nonnegative. In agreement, the assumptions of Theorem 3.7 do not hold since $\text{index}_0(A) = 2$.

The failure of eventual nonnegativity to force eventual exponential nonnegativity observed in the above example can occur even if A is irreducible, as the following example shows.

3.11. Consider the matrix

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \end{bmatrix}$$

and its sequence of powers

$$A^k = \begin{bmatrix} 2^{k-1} & 2^{k-1} & k2^{k-1} & k2^{k-1} \\ 2^{k-1} & 2^{k-1} & k2^{k-1} & k2^{k-1} \\ 0 & 0 & 2^{k-1} & 2^{k-1} \\ 0 & 0 & 2^{k-1} & 2^{k-1} \end{bmatrix} \quad (k = 2, 3, \ldots).$$

The matrix A is an eventually nonnegative matrix with $k_0 = 2$ and $\text{index}_0(A) = 2$. As the assumptions of Theorem 3.7 do not hold, we may not conclude that A is eventually exponentially nonnegative. The $(2,1)$ block of A^k is 0 for all $k \geq 2$, while the one of A is not and contains negative entries. Thus A is not eventually exponentially nonnegative. Indeed,

$$e^A = \begin{bmatrix} 4.1945 & 3.1945 & 7.3891 & 7.3891 \\ 3.1945 & 4.1945 & 7.3891 & 7.3891 \\ -1 & 1 & 4.1945 & 3.1945 \\ 1 & -1 & 3.1945 & 4.1945 \end{bmatrix}, \quad e^{3A} = \begin{bmatrix} 202.2 & 201.2 & 1210.3 & 1210.3 \\ 201.2 & 202.2 & 1210.3 & 1210.3 \\ -3 & 3 & 202.2 & 201.2 \\ 3 & -3 & 201.2 & 202.2 \end{bmatrix}.$$

We now turn our attention to necessary conditions for eventual exponential nonnegativity for which we need to quote some results from [11]. Note that in the first theorem below from [11], we have added the assumption that A is not nilpotent; the need for this assumption is observed in [5].

THEOREM 3.12 (see [11, Theorem 2.3]). $A \in \mathbb{R}^{n\times n}$ A A^T

THEOREM 3.13 (see [11, Theorem 2.4]). $A \in \mathbb{R}^{n\times n}$ A^T $\rho(A)$ A

$$\lim_{k\to\infty} \left(\frac{A}{\rho(A)}\right)^k = xy^T,$$

x y A $\rho(A)$ $x^T y = 1$

THEOREM 3.14. $A \in \mathbb{R}^{n\times n}$

(i) e^A e^{A^T}

(ii) $\rho(e^A)$ e^A $\rho(e^A) = e^{\rho(A)}$ $a_0 \geq 0$ $\lim_{k\to\infty}((A + aI)/(\rho(A + aI))^k = xy^T$ $a > a_0$ x y A $\rho(A)$ $x^T y = 1$

(i) Let A be eventually exponentially nonnegative. As $(e^A)^k = e^{kA}$, it follows that e^A is eventually nonnegative. Thus, by Theorem 3.12 and since e^A and e^{A^T} are not nilpotent, they have the Perron–Frobenius property.

(ii) From (i) we specifically have that $\rho(e^A) \in \sigma(e^A)$. Let x, y be right and left nonnegative eigenvectors, respectively, corresponding to $\rho(e^A)$ and normalized so that $x^T y = 1$. As in the proof of Theorem 3.3, $\rho(e^A) = e^\lambda$ for some $\lambda \in \sigma(A)$ with $\lambda > \operatorname{Re}\mu$ for all $\mu \in \sigma(A) \setminus \{\lambda\}$. This means that there exists large enough $a_0 > 0$, such that for all $a \geq a_0$,

$$\rho(A + aI) = \lambda + a > |\mu + a| \quad \text{for all } \mu \in \sigma(A), \ \ \mu \neq \lambda.$$

As $A + aI$ and e^A share eigenvectors, we obtain that for all $a > a_0$, $A + aI$ and $A^T + aI$ both have the Perron–Frobenius property, with $\lambda + a$ being simple and their only dominant eigenvalue. Applying Theorem 3.13 to $A + aI$, we thus obtain

$$\lim_{k\to\infty} \frac{1}{\rho(A+aI)^k}(A + aI)^k = xy^T \geq 0. \qquad \square \tag{3.2}$$

. 3.15. Referring to the proof of Theorem 3.14, by (3.2) we have that if $(xy^T)_{ij} > 0$, then $\left((A + aI)^k\right)_{ij} > 0$ for all k sufficiently large. In particular, if $xy^T > 0$, then $A + aI$ is eventually nonnegative for all $a > a_0$. If, however, xy^T is nonnegative but not strictly positive, $A + aI$ can fail to be eventually nonnegative for all $a \in \mathbb{R}$. This situation is illustrated by the matrix A in Example 3.11.

4. Points of nonnegative potential. In this section $A \in \mathbb{R}^{n\times n}$ denotes an eventually exponentially nonnegative matrix with exponential index $t_0 = t_0(A) \geq 0$. We will study points of nonnegative potential, that is, the set

$$X_A(\mathbb{R}^n_+) \ = \ \{x_0 \in \mathbb{R}^n \mid (\exists \hat{t} = \hat{t}(x_0) \geq 0)\,(\forall t \geq \hat{t})\,[e^{tA}x_0 \geq 0]\}. \tag{4.1}$$

$X_A(\mathbb{R}^n_+)$ comprises all initial points giving rise to trajectories of (1.1) that reach $\mathbb{R}^n_+$ at some finite time and stay in $\mathbb{R}^n_+$ for all time thereafter.

First, let us recall some basic facts and terminology on convex cones in $\mathbb{R}^n$. Our references are [1, Chapter 1] and [12]. A convex set $K \subseteq \mathbb{R}^n$ is called a , if $aK \subseteq K$ for all $a \geq 0$. A convex cone is called , if it consists of all finite nonnegative linear combinations of the elements of a finite set. A convex cone K is , if $K \cap (-K) = \{0\}$ and , if its topological interior is nonempty. A pointed, solid convex cone is called a , . The nonnegative orthant $\mathbb{R}^n_+$ is indeed a proper cone; it is also a polyhedral cone, comprising all finite nonnegative combinations of the standard basis vectors. Any subset of $\mathbb{R}^n$ of the form $K = S\mathbb{R}^n_+$, where S is an invertible matrix, is a proper polyhedral cone and referred to as a .

Given an eventually exponentially nonnegative matrix $A \in \mathbb{R}^{n\times n}$ with exponential index $t_0 = t_0(A) \geq 0$, define the simplicial cone

$$K = e^{t_0 A}\mathbb{R}^n_+ = \{x_0 \in \mathbb{R}^n \mid (\exists y \geq 0)\,[x_0 = e^{t_0 A}y]\}$$

and consider the sets

$$Y_A(K) \ = \ \{x_0 \in \mathbb{R}^n \mid (\exists \hat{t} = \hat{t}(x_0) \geq 0)\,[e^{\hat{t}A}x_0 \in K]\} \tag{4.2}$$

and

$$X_A(K) \ = \ \{x_0 \in \mathbb{R}^n \mid (\exists \hat{t} = \hat{t}(x_0) \geq 0)\,(\forall t \geq \hat{t})\,[e^{tA}x_0 \in K]\}. \tag{4.3}$$

LEMMA 4.1. . K, $Y_A(K)$ $K \subseteq \mathbb{R}^n_+ \subseteq Y_A(K)$

. We have that $K \subseteq \mathbb{R}^n_+$ since $e^{t_0 A} \geq 0$. If $x_0 \in \mathbb{R}^n_+$, then for $\hat{t} = 2t_0$, $e^{\hat{t}A}x_0 = e^{t_0 A}\,(e^{t_0 A}x_0) \in K$. Hence, $\mathbb{R}^n_+ \subseteq Y_A(K)$. □

Note that the sets $Y_A(K)$, $X_A(K)$, and $X_A(\mathbb{R}^n_+)$ are convex cones. They are not necessarily closed sets, however. For example, when

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},$$

it can be shown that $X_A(\mathbb{R}^2_+)$ consists of the whole upper plane excluding the negative x-axis.

The set $Y_A(K)$ comprises initial points for which the trajectories enter K at some time. The set $X_A(K)$ comprises initial points for which the trajectories enter K at some time and remain in K for all time thereafter. The set of points of nonnegative potential, $X_A(\mathbb{R}^n_+)$, comprises initial points for which the trajectories at some time become nonnegative and remain nonnegative for all time thereafter. Next we shall argue that $Y_A(K)$, $X_A(K)$, and $X_A(\mathbb{R}^n_+)$ coincide and interpret this result subsequently.

PROPOSITION 4.2. . $A \in \mathbb{R}^{n\times n}$. $t_0 = t_0(A) \geq 0$. $K = e^{t_0 A}\mathbb{R}^n_+$

$$Y_A(K) \;=\; X_A(\mathbb{R}^n_+) \;=\; X_A(K).$$

. We begin by proving the first equality. If $x_0 \in Y_A(K)$, then there exists $\hat{t} \geq 0$ and $y \geq 0$ such that $e^{\hat{t}A}x_0 = e^{t_0 A}y$. Thus, $x_0 = e^{(t_0-\hat{t})A}y$ and so $e^{tA}x_0 = e^{(t+t_0-\hat{t})A}y \geq 0$ if $t + t_0 - \hat{t} \geq t_0$, i.e., for all $t \geq \hat{t}$. It follows that $x_0 \in X_A(\mathbb{R}^n_+)$, i.e., $Y_A(K) \subseteq X_A(\mathbb{R}^n_+)$. For the opposite containment, let $x_0 \in X_A(\mathbb{R}^n_+)$; that is, there exists $\hat{t} \geq 0$ such that $e^{tA}x_0 \geq 0$ for all $t \geq \hat{t}$. Let $\tilde{t} = \hat{t} + t_0$. Then $e^{\tilde{t}A}x_0 = e^{t_0 A}(e^{\hat{t}Ax_0}) \in K$, proving that $X_A(\mathbb{R}^n_+) \subseteq Y_A(K)$ and thus equality holds.

For the second equality, clearly $X_A(K) \subseteq X_A(\mathbb{R}^n_+)$ since $K \subseteq \mathbb{R}^n_+$. To show the opposite containment, let $x_0 \in X_A(\mathbb{R}^n_+)$. Then there exists $\hat{t} \geq 0$ such that $e^{t_0 A}e^{sA}x_0 \in K$ for all $s \geq \hat{t}$. That is, $e^{tA}x_0 \in K$ for all $t \geq t_0 + \hat{t}$ and thus $x_0 \in X_A(K)$. □

. 4.3. Referring to Proposition 4.2, we must make the following observations:

(i) If $t_0 = 0$ (i.e., if $A \overset{\mathrm{s}}{\geq} 0$, or equivalently if $e^{tA} \geq 0$ for all $t \geq 0$), then $K = \mathbb{R}^n_+$. In this case, $X_A(\mathbb{R}^n_+)$ coincides with the . of the nonnegative orthant for an essentially nonnegative matrix, which is studied in detail in [10, 9].

(ii) The equality $X_A(\mathbb{R}^n_+) = X_A(K)$, in conjunction with Lemma 4.1, can be interpreted as saying that the simplicial cone $K = e^{t_0 A}\mathbb{R}^n_+$ serves as an attractor set for trajectories emanating at points of nonnegative potential; in other words, trajectories emanating in $X_A(\mathbb{R}^n_+)$ always reach and remain in $K \subseteq \mathbb{R}^n_+$ after a finite time.

(iii) Our observations so far imply that the trajectory emanating from a point of nonnegative potential will enter cone K; however, it may subsequently exit K while it remains nonnegative, and it will eventually re-enter K and remain in K for all finite time thereafter. This situation is illustrated by the following example.

. 4.4. Consider the matrix

$$A = \begin{bmatrix} 0.3929 & -0.8393 & 1.1071 & 1.3393 \\ 1.0357 & 0.6964 & -0.5357 & 0.8036 \\ 1.0357 & -0.3036 & 0.4643 & 0.8036 \\ 1.4643 & 1.0536 & -0.9643 & 0.4464 \end{bmatrix}.$$

It can be checked that A and A^T have the strong Perron–Frobenius property and so, by Theorems 3.2 and 3.3, A is an eventually exponentially positive matrix. Using MATLAB and a bisection method, we estimated (within five decimals) the exponential index to be $t_0 = t_0(A) = 2.64378$. The matrices e^A and $e^{t_0 A}$ are

$$e^A = \begin{bmatrix} 3.6277 & -0.7991 & 1.4260 & 3.1345 \\ 3.0341 & 2.2579 & -0.6987 & 2.7958 \\ 3.0341 & -0.4604 & 2.0196 & 2.7958 \\ 3.3050 & 1.4836 & -0.9696 & 3.5701 \end{bmatrix}$$

and

$$e^{t_0 A} = \begin{bmatrix} 91.902 & 3.5982 & 14.0615 & 88.299 \\ 91.499 & 18.162 & 0.3981 & 87.801 \\ 91.499 & 4.0959 & 14.4643 & 87.801 \\ 91.897 & 17.494 & 0 & 88.469 \end{bmatrix}.$$

Hence the cone $K = e^{t_0 A}\mathbb{R}^n_+$ is the cone generated by the columns of the matrix $e^{t_0 A}$ above. Consider now the following trajectory points $x(t) = e^{tA}x(0)$:

$$x_0 = x(0) = \begin{bmatrix} -1.1617 \\ 0.6014 \\ 0.9693 \\ 1.0887 \end{bmatrix}, \quad x(1) = e^A x_0 = \begin{bmatrix} 0.1 \\ 0.2 \\ 1.2 \\ 0 \end{bmatrix}, \quad x(2) = e^{2A}x_0 = \begin{bmatrix} 1.9141 \\ -0.0834 \\ 2.6348 \\ -0.5363 \end{bmatrix},$$

$$e^{(t_0+1)A}x_0 = \begin{bmatrix} 26.7836 \\ 13.2600 \\ 27.3263 \\ 12.6884 \end{bmatrix}, \quad e^{(t_0+2)A}x_0 = \begin{bmatrix} 165.3049 \\ 127.5845 \\ 165.8206 \\ 126.9949 \end{bmatrix}, \quad e^{(2t_0+1)A}x_0 = \begin{bmatrix} 4013.8 \\ 3816.4 \\ 4014.3 \\ 3815.8 \end{bmatrix}.$$

Observe the following: $e^{(t_0+1)A}x_0 \in K$ since $e^A x_0 \in \mathbb{R}^n_+$; $e^{(t_0+2)A}x_0 \notin K$ since $e^{2A}x_0 \notin \mathbb{R}^n_+$; $e^{(2t_0+1)A}x_0 \in K$ since $e^{(t_0+1)A}x_0 \in \mathbb{R}^n_+$; finally, trajectory points $x(t)$ are in K for all $t \geq 2t_0 + 1$. In other words, the trajectory emanating at x_0 enters K, exits K, and eventually re-enters and remains in K for all time thereafter.

In view of the above example, a natural question arises: When is it possible that all trajectories emanating in $X_A(\mathbb{R}^n_+)$ reach and never exit K? This is equivalent to asking whether or not $e^{tA}K \subseteq K$ for all $t \geq 0$. To resolve this question, we will invoke the following extension of Lemma 3.1 from $\mathbb{R}^n_+$ to simplicial cones, which can be found in [13, 14].

LEMMA 4.5. [illegible] $A \in \mathbb{R}^{n\times n}$ [illegible] $K = S\mathbb{R}^n_+$ [illegible] $S \in \mathbb{R}^{n\times n}$ [illegible] $a \geq 0$ [illegible] $(A + aI)K \subseteq K$ [illegible] $e^{tA}K \subseteq K$ [illegible] $t \geq 0$

[illegible]. Consider the similarity transformation $A \to B = S^{-1}AS$. We claim that there exists $a \geq 0$ such that $(A + aI)K \subseteq K$ if and only if $B \overset{\text{s}}{\geq} 0$. Indeed, if $(A + aI)K \subseteq K$, then

$$(B + aI)\mathbb{R}^n_+ = S^{-1}(A + aI)S\mathbb{R}^n_+ = S^{-1}(A + aI)K \subseteq S^{-1}K = \mathbb{R}^n_+.$$

Conversely, if $B \overset{\text{s}}{\geq} 0$, then there exists $a \geq 0$ such $B + aI = S^{-1}(A + aI)S \geq 0$. Hence for each $x \in K$, there exists $y \geq 0$ such that

$$S^{-1}(A + aI)x = S^{-1}(A + aI)Sy = z \geq 0.$$

That is, $(A + aI)Sy = Sz \in K$. Similarly, one can show that $e^{tA}K \subseteq K$ for all $t \geq 0$ if and only if $e^{tB} \geq 0$ for all $t \geq 0$. □

We note in passing that Lemma 4.5 holds more generally for every polyhedral cone K; see [13, 14].

COROLLARY 4.6. [illegible] $A \in \mathbb{R}^{n\times n}$ [illegible] $t_0 = t_0(A) \geq 0$ [illegible] $K = e^{t_0 A}\mathbb{R}^n_+$ [illegible] $e^{tA}K \subseteq K$ [illegible] $t \geq 0$ [illegible] $t_0 = 0$ ([illegible] $A \overset{s}{\geq} 0$)

[illegible]. If $t_0 = 0$, then $K = \mathbb{R}^n_+$ and $e^{tA} \geq 0$ for all $t \geq 0$. For the converse, suppose $e^{tA}K \subseteq K$ for all $t \geq 0$. We must show that $t_0 = 0$. Let $y \geq 0$ and consider $x_0 = e^{t_0 A}y \in K$. As $e^{tA}x_0 \in K$ for all $t \geq 0$, there must exist $z \geq 0$ such that

$$e^{(t+t_0)A}\, y \;=\; e^{t_0 A}\, z \quad \text{for all } t \geq 0.$$

But this means $e^{tA}y = z \geq 0$ for all $t \geq 0$. Since y was taken arbitrary in $\mathbb{R}^n_+$, we have $e^{tA}\mathbb{R}^n_+ \subseteq \mathbb{R}^n_+$ for all $t \geq 0$; that is, $t_0 = 0$. □

We conclude this section with a discussion on a possible numerical test for points of nonnegative potential. When $A = [a_{ij}] \overset{s}{\geq} 0$, $X_A(\mathbb{R}^n_+)$ admits a numerical characterization reported in [10] and briefly described in the following. Consider the sequence $\{x_k\}$ generated from x_0 by the Cauchy–Euler finite differences scheme

$$x_k = (I + hA)^k x_0, \quad k = 0, 1, \ldots,$$

which we refer to as [illegible] ([illegible] h) [illegible] x_0. Define the quantity

$$h(A) = \sup\left\{ h \mid \min_{1\leq i\leq n} (1 + ha_{ii}) > 0 \right\}$$

and notice that $h(A) = \sup\{h \mid (I + hA) \geq 0\} > 0$, as well as that $h(A) = \infty$ when $A \geq 0$.

For any $h \in (0, h(A))$, denote by $X_{A,h}(\mathbb{R}^n_+)$ the set of all initial states $x_0 \in \mathbb{R}^n$ that give rise to discrete trajectories $\{x_k\}$ which become and remain (due to nonnegativity of $I + hA$) nonnegative; that is,

$$X_{A,h}(\mathbb{R}^n_+) = \{x_0 \in \mathbb{R}^n \mid (\exists k_0 = k_0(x_0) \geq 0)\,(\forall k \geq k_0)\,[(I + hA)^k x_0 \in \mathbb{R}^n_+]\}.$$

We refer to $X_{A,h}(\mathbb{R}^n)$ as the [illegible] ([illegible] $\mathbb{R}^n_+$ [illegible] A [illegible] h). The geometric and algebraic properties of the discrete reachability cone are studied extensively in [8, 10].

THEOREM 4.7 (see [10]). [illegible] $A \in \mathbb{R}^{n\times n}$ [illegible] $h \in (0, h(A))$ [illegible] $(I + hA)$ [illegible] $X_A(\mathbb{R}^n_+) = X_{A,h}(\mathbb{R}^n_+)$

When $A \overset{s}{\geq} 0$, Theorem 4.7 suggests a simple test to find out whether or not a given initial point x_0 belongs to $X_A(\mathbb{R}^n_+)$: 1. Choose a positive $h < h(A)$ such that the iteration matrix $I + hA$ is invertible. 2. Check whether for some nonnegative integer k, $x_k = (I + hA)^k x_0$ is nonnegative (in which case $x_0 \in X_A(\mathbb{R}^n_+)$) or decide that x_k will never be nonnegative (in which case $x_0 \notin X_A(\mathbb{R}^n_+)$).

As noted in [15], Theorem 4.7 can be generalized from $\mathbb{R}^n_+$ to any simplicial cone K such that $e^{tA}K \subseteq K$ for all $t \geq 0$. Thus, in view of Proposition 4.2, the question arising is whether the above test can be extended to $X_A(\mathbb{R}^n_+) = X_A(K)$, when A is eventually exponentially nonnegative with exponential index $t_0 \geq 0$ and $K = e^{t_0 A}\mathbb{R}^n_+$. By Corollary 4.6, however, it follows that the answer is in the negative when

$t_0 > 0$. The development of a characterization of points of nonnegative potential in terms of discrete trajectories will likely require a close examination of the generalized eigenspaces of A as in the proof of Theorem 4.7. We plan to undertake this task in future work, as well as perform a numerical analysis of the associated test.

5. Conclusions. We considered the problem of when a trajectory $x(t) = e^{tA}x_0$ $(t \geq 0)$ becomes and remains nonnegative. Naturally, we needed to study (1) matrices A for which e^{tA} becomes and remains nonnegative and (2) initial points x_0 giving rise to nonnegative trajectories, which we called points of nonnegative potential. The combination of such matrices and initial points results in trajectories that reach and stay in the nonnegative orthant. We discovered that eventual nonnegativity of the exponential matrix is intimately related to eventual nonnegativity of the powers of A (section 3). We also found that the collection of points of nonnegative potential coincides with the collection of initial points that reach and stay in a certain simplicial cone K associated with e^{tA}. Interestingly, trajectories emanating at points of nonnegative potential may enter and subsequently exit this cone K; however, K eventually attracts such trajectories permanently (section 4). Our results generalize and parallel well-known facts in nonnegative systems theory and are illustrated with several examples.

Acknowledgment. The authors would like to sincerely thank the anonymous referees and the editor, Daniel Szyld, for several comments that helped us correct and improve the original manuscript.

REFERENCES

[1] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1994.
[2] A. Berman, M. Neumann, and R. J. Stern, *Nonnegative Matrices in Dynamic Systems*. Wiley-Interscience, New York, 1989.
[3] S. Carnochan Naqvi and J. J. McDonald, *The combinatorial structure of eventually nonnegative matrices*, Electron. J. Linear Algebra, 9 (2002), pp. 255–269.
[4] S. Carnochan Naqvi and J. J. McDonald, *Eventually nonnegative matrices are similar to semi-nonnegative matrices*, Linear Algebra Appl., 381 (2004), pp. 245–258.
[5] A. Elhashash and D. B. Szyld, *Perron-Frobenius Properties of General Matrices*, preprint, 2007.
[6] L. Farina and S. Rinaldi, *Positive Linear Systems: Theory and Applications*, Wiley-Interscience, New York, 2000.
[7] M. Neumann and R. J. Stern, *Boundary results for positively invariant cones and their reachability cones*, Linear Multilinear Algebra, 17 (1985), pp. 143–154.
[8] M. Neumann and R. J. Stern, *Cone reachability for linear differential systems*, Appl. Anal., 20 (1986), pp. 57–71.
[9] M. Neumann and M. Tsatsomeros, *Symbiosis points for linear differential systems*, Linear Multilinear Algebra, 30 (1991), pp. 49–59.
[10] M. Neumann, R. J. Stern, and M. Tsatsomeros, *The reachability cones of essentially nonnegative matrices*, Linear Multilinear Algebra, 28 (1991), pp. 213–224.
[11] D. Noutsos, *On Perron-Frobenius property of matrices having some negative entries*, Linear Algebra Appl., 412 (2005), pp. 132–153.
[12] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1997.
[13] H. Schneider and M. Vidyasagar, *Cross-positive matrices*, SIAM J. Numer. Anal., 7 (1970), pp. 508–519.
[14] R. J. Stern and M. J. Tsatsomeros, *Extended M-matrices and subtangentiality*, Linear Algebra Appl., 97 (1987), pp. 1–11.
[15] M. J. Tsatsomeros, *Reachability of Nonnegative and Symbiotic States*, Ph.D. thesis, University of Connecticut, Storrs, CT, 1990.
[16] B. G. Zaslavsky, *Eventually nonnegative realization of difference control systems*, in Dynamical Systems and Related Topics, Adv. Ser. Dynam. Systems 9, World Scientific, River Edge, NJ, 1991, pp. 573–602.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 713–730

NONNEGATIVE MATRIX FACTORIZATION BASED ON ALTERNATING NONNEGATIVITY CONSTRAINED LEAST SQUARES AND ACTIVE SET METHOD*

HYUNSOO KIM† AND HAESUN PARK†

Abstract. Nonnegative matrix factorization (NMF) determines a lower rank approximation of a matrix $A \in \mathbb{R}^{m\times n} \approx WH$ where an integer $k \ll \min(m,n)$ is given and nonnegativity is imposed on all components of the factors $W \in \mathbb{R}^{m\times k}$ and $H \in \mathbb{R}^{k\times n}$. NMF has attracted much attention for over a decade and has been successfully applied to numerous data analysis problems. In applications where the components of the data are necessarily nonnegative, such as chemical concentrations in experimental results or pixels in digital images, NMF provides a more relevant interpretation of the results since it gives nonsubtractive combinations of nonnegative basis vectors. In this paper, we introduce an algorithm for NMF based on alternating nonnegativity constrained least squares (NMF/ANLS) and the active set–based fast algorithm for nonnegativity constrained least squares with multiple right-hand side vectors, and we discuss its convergence properties and a rigorous convergence criterion based on the Karush–Kuhn–Tucker (KKT) conditions. In addition, we also describe algorithms for sparse NMFs and regularized NMF. We show how we impose a sparsity constraint on one of the factors by L_1-norm minimization and discuss its convergence properties. Our algorithms are compared to other commonly used NMF algorithms in the literature on several test data sets in terms of their convergence behavior.

Key words. nonnegative matrix factorization, lower rank approximation, two-block coordinate descent method, Karush–Kuhn–Tucker (KKT) conditions, nonnegativity constrained least squares, active set method

AMS subject classification. 15A23

DOI. 10.1137/07069239X

1. Introduction.

Given a nonnegative matrix $A \in \mathbb{R}^{m\times n}$ and a desired rank $k \ll \min(m,n)$, nonnegative matrix factorization (NMF) searches for nonnegative factors W and H that give a lower rank approximation of A as

$$A \approx WH \quad \text{s.t.} \quad W, H \geq 0, \tag{1.1}$$

where $W, H \geq 0$ means that all elements of W and H are nonnegative. The problem in (1.1) is commonly reformulated as the following optimization problem:

$$\min_{W,H} f(W,H) \equiv \frac{1}{2}\|A - WH\|_F^2 \quad \text{s.t.} \quad W, H \geq 0, \tag{1.2}$$

where $W \in \mathbb{R}^{m\times k}$ is a basis matrix and $H \in \mathbb{R}^{k\times n}$ is a coefficient matrix. In many data analysis problems, typically each column of A corresponds to a data point in the m-dimensional space.

NMF may give a simple interpretation due to nonsubtractive combinations of nonnegative basis vectors and has recently received much attention. Applications of

*Received by the editors June 21, 2007; accepted for publication (in revised form) by L. Reichel January 3, 2008; published electronically July 2, 2008. This material is based upon work supported in part by the National Science Foundation through grants CCF-0621889 and CCF-0732318. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

http://www.siam.org/journals/simax/30-2/69239.html

†College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332 (hskim@cc.gatech.edu, hpark@cc.gatech.edu).

NMF are numerous, including image processing [21], text data mining [31], subsystem identification [19], and cancer class discovery [4, 8, 18]. It has been over a decade since NMF was first proposed by Paatero and Tapper [27] (in fact, as positive matrix factorization) in 1994. Various types of NMF techniques have been proposed in the literature [5, 13, 25, 32, 34], which include Lee and Seung's popular iterative multiplicative update algorithms [21, 22], gradient descent methods [24], and alternating least squares [1]. Paatero and Tapper [27] originally proposed an algorithm for NMF using a constrained alternating least squares algorithm to solve (1.2). Unfortunately, this approach has not obtained wide attention especially after Lee and Seung's multiplicative update algorithm was proposed [1, 24]. The main difficulty was extremely slow speed caused by a vast amount of hidden redundant computation related to satisfying the nonnegativity constraints exactly. One may try to deal with the nonnegativity constraints in an approximate sense for faster algorithm. However, we will show that it is important to satisfy the constraints exactly for the overall convergence of the algorithm and that this property provides very practical and faster algorithm as well. In addition, faster algorithms that exactly satisfy the nonnegativity constraints in the least squares with multiple right-hand sides already exist [27, 36], which we will discuss and utilize in our proposed NMF algorithms.

In this paper, we provide a framework of the two-block coordinate descent method for NMF. This framework provides a convenient way to explain and compare most of the existing commonly used NMF algorithms and to discuss their convergence properties. We then introduce an NMF algorithm which is based on alternating nonnegativity constrained least squares (NMF/ANLS) and the active set method. Although many existing NMF algorithms produce the factors which are often sparse, the formulation of the NMF shown in (1.2) does not guarantee the sparsity in the factors. We introduce an NMF formulation and algorithm that imposes sparsity constraint on one of the factors by L_1-norm minimization and discuss its convergence properties. The L_1-norm minimization term is formulated in such a way that the proposed sparse NMF algorithm also fits into the framework of the two-block coordinate descent method and accordingly its convergence properties become easy to understand.

The rest of this paper is organized as follows. We present the framework of the two-block coordinate descent method and provide a brief overview of various existing NMF algorithms in section 2. In section 3, we introduce our NMF/ANLS algorithm and discuss its convergence properties. In section 4, we describe some variations of the NMF/ANLS algorithm, which include the method designed to impose sparsity on one of the factors through the addition of an L_1-norm minimization term in the problem formulation. Our algorithms are compared to other commonly used NMF algorithms in the literature on several test data sets in section 6. Finally, summary and discussion are given in section 7.

2. A two-block coordinate descent framework for NMF algorithms and convergence properties. In most of the currently existing algorithms for NMF, the basic framework is to reformulate the nonconvex minimization problem shown in (1.2) as a two-block coordinate descent problem [2]. Given a nonnegative matrix $A \in \mathbb{R}^{m \times n}$ and an integer $k < \min(m, n)$, one of the factors, say $H \in \mathbb{R}^{k \times n}$, is initialized with nonnegative values. Then one may iterate the following alternating nonnegativity constrained least squares (ANLS) until a convergence criterion is satisfied:

$$\min_{W \geq 0} \|H^T W^T - A^T\|_F^2, \tag{2.1}$$

where H is fixed, and

$$\min_{H \geq 0} \|WH - A\|_F^2, \tag{2.2}$$

where W is fixed. Alternatively, after initializing W, one may iterate (2.2) and then (2.1) until a convergence criterion is satisfied. Each subproblem shown in (2.1)–(2.2) can be solved by projected quasi-Newton optimization [37, 15], projected gradient descent optimization [24], or nonnegativity constrained least squares [27, 16, 28].

Note that the original NMF problem of (1.2) is nonconvex, and most nonconvex optimization algorithms guarantee only the stationarity of limit points. Since the problem formulation is symmetric with respect to initialization of the factors H or W, for simplicity of discussion, we will assume that the iteration is performed with the initialization of the factor H. Then the above iteration can be expressed as follows:

- Initialize H with a nonnegative matrix $H^{(0)}$; $t \leftarrow 0$.
- Repeat until a stopping criterion is satisfied:
  - $W^{(t+1)} = \arg\min_W f(W, H^{(t)})$ s.t. $W \geq 0$.
  - $H^{(t+1)} = \arg\min_H f(W^{(t+1)}, H)$ s.t. $H \geq 0$.
  - $t \leftarrow t + 1$.

According to the Karush–Kuhn–Tucker (KKT) optimality conditions, (W, H) is a stationary point of (1.2) if and only if

$$\begin{aligned} &W \geq 0, && H \geq 0, \\ &\nabla_W f(W,H) = WHH^T - AH^T \geq 0, && \nabla_H f(W,H) = W^TWH - W^TA \geq 0, \\ &W. * \nabla_W f(W,H) = 0, && H. * \nabla_H f(W,H) = 0, \end{aligned} \tag{2.3}$$

where $.*$ denotes componentwise multiplication [11].

For (1.2), when the block coordinate descent algorithm is applied, then no matter how many subblocks into which the problem is partitioned, if the subproblems have unique solutions, then the limit point of the sequence is a stationary point [2]. For two-block problems, Grippo and Siandrone [12] presented a stronger result. The result does not require uniqueness of the solution in each subproblem, which is that any limit point of the sequence generated based on the optimal solutions of each of the two subblocks is a stationary point. Since the subproblems (2.1) and (2.2) are convex but not strongly convex, they do not necessarily have unique solutions. However, according to the two-block result, it is still the case that any limit point will be a stationary point. We emphasize that for convergence to a stationary point, it is important to find an optimal solution for each subproblem.

In one of the most commonly utilized NMF algorithms due to Lee and Seung [21, 22], NMF is computed using the following norm-based multiplicative update rules (NMF/NUR) of W and H, which is a variation of the gradient descent method:

$$W_{iq} \leftarrow W_{iq} \frac{(AH^T)_{iq}}{(W(HH^T))_{iq}} \tag{2.4}$$

for $1 \leq i \leq m$ and $1 \leq q \leq k$,

$$H_{qj} \leftarrow H_{qj} \frac{(W^TA)_{qj}}{((W^TW)H)_{qj}} \tag{2.5}$$

for $1 \leq q \leq k$ and $1 \leq j \leq n$. Each iteration may in fact break down since the denominators in both (2.4) and (2.5) can be zeros. Accordingly, in practical algorithms, a small positive number is added to each denominator to prevent division by zero. There are several variations of NMF/NUR [8, 30, 6].

Lee and Seung also designed an NMF algorithm using the divergence-based multiplicative update rules (NMF/DUR) [22] to minimize the divergence:

$$D(A||WH) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(A_{ij}\ln\frac{A_{ij}}{(WH)_{ij}} - A_{ij} + (WH)_{ij}\right) \quad \text{s.t.} \quad W, H \geq 0. \tag{2.6}$$

Strictly speaking, this formulation is not a bound constrained problem, which requires the objective function to be well-defined at any point of the bounded region, since the log function is not well-defined if $A_{ij} = 0$ or $(WH)_{ij} = 0$ [24]. The divergence is also nonincreasing during iterations. Gonzales and Zhang [11] claimed that these nonincreasing properties of multiplicative update rules may not imply the convergence to a stationary point within a realistic amount of run time for problems of meaningful sizes. Lin [24] devised an NMF algorithm based on projected gradient methods. However, it is known that gradient descent methods may suffer from slow convergence due to a possible zigzag phenomenon.

Berry et al. [1] proposed an NMF algorithm based on alternating least squares (NMF/ALS). This algorithm computes the solutions to the subproblems (2.1) and (2.2) as an unconstrained least squares problems with multiple right-hand sides and sets negative values in the solutions W and H to zero during iterations to enforce nonnegativity. Although this may give a faster algorithm for approximating each subproblem, the convergence of the overall algorithm is difficult to analyze since the subproblems are formulated as constrained least squares problems but the solutions are not those of the constrained least squares.

Zdunek and Cichocki [37] developed a quasi-Newton optimization approach with projection. In this algorithm, the negative values of W and H are replaced with a very small positive value. Again, setting negative values to zero or small positive values for imposing nonnegativity makes theoretical analysis of the convergence of the algorithm difficult [3]. The projection step can increase the objective function value and may lead to nonmonotonic changes in the objective function value resulting in inaccurate approximations.

A more detailed review of NMF algorithms can be found in [1].

3. NMF based on alternating nonnegativity constrained least squares and the active set method. In this section, we describe our NMF algorithm based on alternating nonnegativity constrained least squares (NMF/ANLS) that satisfies the nonnegativity constraints in each of the subproblems in (2.1) and (2.2) exactly and therefore has the convergence property that every limit point is a stationary point.

The structures of the two nonnegativity constrained least squares (NLS) problems with multiple right-hand sides shown in (2.1) and (2.2) are essentially the same, and therefore we will concentrate on a general form of the NLS with multiple right-hand sides

$$\min_{G \geq 0} \|BG - Y\|_F^2, \tag{3.1}$$

where $B \in \mathbb{R}^{p \times q}$ and $Y \in \mathbb{R}^{p \times l}$ are given, which can be decoupled into l independent

NLS problems each with single right-hand side as

$$\min_{G\geq 0}\|BG-Y\|_F^2 \rightarrow \min_{\mathbf{g}_1\geq 0}\|B\mathbf{g}_1-\mathbf{y}_1\|_2^2, \quad \ldots \quad ,\min_{\mathbf{g}_l\geq 0}\|B\mathbf{g}_l-\mathbf{y}_l\|_2^2, \tag{3.2}$$

where $G=[\mathbf{g}_1,\ldots,\mathbf{g}_l]\in\mathbb{R}^{q\times l}$ and $Y=[\mathbf{y}_1,\ldots,\mathbf{y}_l]\in\mathbb{R}^{p\times l}$. This objective function is not strictly convex so that it does not ensure a unique solution unless B is full column rank. In the context of the NMF computation, we implicitly assume that the fixed matrices H^T and W involved in (2.1) and (2.2) are of full column rank since they are interpreted as basis matrices for A^T and A, respectively. Each of the NLS problems with single right-hand side vector

$$\min_{\mathbf{g}_j\geq 0}\|B\mathbf{g}_j-\mathbf{y}_j\|_2 \tag{3.3}$$

for $1\leq j\leq l$ can be solved by using the active set method of Lawson and Hanson [20], which is implemented in MATLAB [26] as the function [illegible]. The algorithm is summarized in Algorithm 1.

The following theorem states the necessary and sufficient conditions for a vector g to be a solution for the problem NLS.

THEOREM 1 (Kuhn–Tucker conditions for problem NLS). [illegible] $g\in\mathbb{R}^{n\times 1}$ [illegible]

$$\min\|Bg-b\| \quad s.t. \quad g\geq 0 \tag{3.4}$$

[illegible] $r\in\mathbb{R}^{m\times 1}$ [illegible] 1 [illegible] m [illegible] $\mathcal{E}$ [illegible] $\mathcal{S}$ [illegible] $r=B^T(Bg-y)$

$$g_i=0 \quad i\in\mathcal{E}, \qquad g_i>0 \quad i\in\mathcal{S}, \tag{3.5}$$

$$r_i\geq 0 \quad i\in\mathcal{E}, \qquad r_i=0 \quad i\in\mathcal{S}. \tag{3.6}$$

On termination of Algorithm NLS, the solution vector g satisfies

$$g_i>0, \quad i\in\mathcal{S} \quad \text{and} \quad g_i=0, \quad i\in\mathcal{E} \tag{3.7}$$

and is a solution vector for the [illegible] least squares problem

$$\min_x\|B_{\mathcal{S}}g-y\|_2. \tag{3.8}$$

The dual vector $w=-r=B^T(y-Bg)$ satisfies

$$w_i=0 \quad i\in\mathcal{S} \quad \text{and} \quad w_i\leq 0 \quad j\in\mathcal{E}. \tag{3.9}$$

To enhance the computational speed in solving (3.1) based on Algorithm NLS, we utilize the fast algorithms by Bro and de Jong [3] and van Benthem and Keenan [36]. Bro and de Jong [3] made a substantial speed improvement for solving (3.1) which has multiple right-hand side vectors over a naive application of Algorithm NLS, which is for a single right-hand side problem, by precomputing cross-product terms that appear in the normal equations of the unconstrained least squares problems. Van Benthem and Keenan [36] devised an algorithm that further improves the performance of NLS for multivariate data by initializing the active set $\mathcal{E}$ based on the result from the unconstrained least squares solution and reorganizing the calculations to take advantage of the combinatorial nature of the active set–based solution methods for the NLS with [illegible] right-hand sides.

Algorithm 1 NLS: This algorithm computes the solution for the problem $\min_{g \geq 0} \|Bg - y\|_2$ by the active set method, where $B \in \mathbf{R}^{m \times n}$ and $y \in \mathbf{R}^{m \times 1}$ are given.

Initialization:
$g := 0$
$\mathcal{E} := \{1, 2, \ldots, n\}$ % Initially all indices belong to Active set since $g := 0$
$\mathcal{S} := \phi$ % Initially Passive set is empty
$w := B^T(b - Bg)$.

Do While ($\mathcal{E} \neq \phi$ and $\exists\ j \in \mathcal{E}$ such that $w_j > 0$)

1. Find an index $t \in \mathcal{E}$ such that $w_t = \max\{w_j : j \in \mathcal{E}\}$ % t is the column index of B that can potentially reduce the objective function value by maximum when brought into the Passive set.
2. Move the index t from set $\mathcal{E}$ to set $\mathcal{S}$.
3. Let $B_{\mathcal{S}}$ denote the $m \times n$ matrix defined by
   Column j of $B_{\mathcal{S}} := \begin{cases} \text{column } j \text{ of } B & \text{if } j \in \mathcal{S}, \\ 0 & \text{if } j \in \mathcal{E}. \end{cases}$
   Solve $\min_z \|B_{\mathcal{S}} z - b\|_2$. (% Only the components z_j, $j \in \mathcal{S}$, are determined by this problem.)
   $z_j := 0$ for $j \in \mathcal{E}$.
4. **Do While** ($z_j \leq 0$ for any $j \in \mathcal{S}$)
   (a) Find an index $q \in \mathcal{S}$ such that $g_q/(g_q - z_q) = \min\{g_j/(g_j - z_j) :\ z_j \leq 0, j \in \mathcal{S}\}$.
   (b) $\alpha := g_q/(g_q - z_q)$.
   (c) $g := g + \alpha(z - g)$.
   (d) Move from set $\mathcal{S}$ to set $\mathcal{E}$ all indices $j \in \mathcal{S}$ for which $g_j = 0$.
   (e) Define $B_{\mathcal{S}}$ as in step 3 and
   Solve $\min_z \|B_{\mathcal{S}} z - b\|_2$.
5. **End While** (% $z_j > 0$ for all $j \in \mathcal{S}$)
6. $g := z$.
7. $w := B^T(b - Bg)$.

End While (% $\mathcal{E}$ is empty (all indices are passive) or $w_j \leq 0$ for all $j \in \mathcal{E}$ (objective function value cannot be reduced anymore))

To illustrate the situation in a simpler context, let us for now assume that there is no nonnegativity constraints in the least squares problems shown in (3.2) and (3.3). Then, since an optimal solution $\mathbf{g}_j^*$ for $\min_{\mathbf{g}_j} \|B\mathbf{g}_j - \mathbf{y}_j\|_2$ is $B^\dagger \mathbf{y}_j$ for $j = 1, \ldots, l$, the pseudoinverse $B^\dagger$ of B [9] needs to be computed [illegible] (in fact, we do not recommend forming the pseudoinverse explicitly and it is used here only for explanation). Clearly, it would be extremely inefficient if we treat each subproblem independently and process the matrix B each time. In the case of NLS with multiple right-hand side vectors, the scenario is not this simple since the active set $\mathcal{E}$ may differ in each iteration and for each right-hand side vector, and a solution is obtained based on a subset of columns of the matrix B that corresponds to the passive set in each iteration, as shown in step 3 of Algorithm NLS. However, much of the computation which is potentially redundant in each iteration can be identified and precomputed only once. For example, if the matrix B has full column rank, then by precomputing B^TB and B^TY only once and extracting the necessary components from these for

each passive set, one can obtain the solution efficiently by extracting the normal equations for each passive set avoiding redundant computations [3]. In addition, for the multiple right-hand side case, the computations can be rearranged to be column parallel; i.e., the passive set columns in each step of the active set iteration for all right-hand side vectors are identified collectively at once. Thus, larger sets of common passive sets can be found and more redundant computations can be avoided. More detailed explanations of this algorithm can be found in [36].

As we stated earlier, with the above-mentioned solution method NMF/ANLS, which satisfies the nonnegativity constraint exactly, any limit point will be a stationary point [2, 12]. Lin [24] also discussed the convergence properties of alternating nonnegativity constrained least squares and showed that any limit point of the sequence (W, H) generated by alternating nonnegativity constrained least squares is a stationary point of (1.2) when the objective function is convex, and not necessarily strictly convex. The NMF is clearly not unique since there exist nonsingular matrices $X \in \mathbb{R}^{k\times k}$ including scaling and permutation matrices satisfying $WX \geq 0$ and $X^{-1}H \geq 0$, and these factors give $\|A - WH\|_F = \|A - WXX^{-1}H\|_F$. To provide a fair comparison among the computed factors based on various algorithms in the presence of this nonuniqueness, after convergence, the columns of the basis matrix W are often normalized to unit L_2-norm and the rows of H are adjusted so that the objective function value is not changed. However, we would like to note that normalizing the computed factors after each iteration makes the convergence results of the two-block coordinate descent method not applicable since the normalization alters the objective function of the subproblems expressed in (2.1) and (2.2).

4. Algorithms for sparse NMF based on alternating nonnegativity constrained least squares. One of the interesting properties of NMF is that it often generates sparse factors that allow us to discover parts-based basis vectors. Although the results presented in [21] show that the computed NMF generated parts-based basis vectors, the generation of a parts-based basis by NMF depends on the data and the algorithm [14, 23]. Several approaches [7, 14, 29, 30] have been proposed to explicitly control the degree of sparseness in the factors of NMF. In this section, we propose algorithms for the sparse NMF that follows the framework of the two-block coordinate descent methods and therefore guarantees that every limit point is a stationary point. In particular, we propose an L_1-norm–based constrained NMF formulation to control the sparsity of one of the factors.

4.1. Constrained NMF based on alternating nonnegativity constrained least squares. Pauca, Piper, and Plemmons [30] proposed the following constrained NMF (CNMF) formulation for the purpose of obtaining a sparse NMF:

$$\min_{W,H} \frac{1}{2}\{\|A - WH\|_F^2 + \alpha\|W\|_F^2 + \beta\|H\|_F^2\} \quad \text{s.t.} \quad W, H \geq 0, \tag{4.1}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are the parameters to be chosen and are supposed to control the sparsity of W and H, respectively. An algorithm was developed based on multiplicative update rules for the CNMF formulation.

We now show how the formulation in (4.1) can be recast into the ANLS framework and developed into an algorithm CNMF/ANLS for which every limit point is a stationary point. The algorithm CNMF/ANLS begins with the initialization of H

with nonnegative values. Then the following ANLS can be iterated:

$$\min_{W\geq 0} \left\| \begin{pmatrix} H^T \\ \sqrt{\alpha} I_k \end{pmatrix} W^T - \begin{pmatrix} A^T \\ 0_{k\times m} \end{pmatrix} \right\|_F^2, \tag{4.2}$$

where I_k is a $k \times k$ identity matrix and $0_{k\times m}$ is a zero matrix of size $k \times m$, and

$$\min_{H\geq 0} \left\| \begin{pmatrix} W \\ \sqrt{\beta} I_k \end{pmatrix} H - \begin{pmatrix} A \\ 0_{k\times n} \end{pmatrix} \right\|_F^2, \tag{4.3}$$

where $0_{k\times n}$ is a zero matrix of size $k \times n$. Similarly, one may initialize $W \in \mathbb{R}^{m\times k}$ and alternate the above in the order of solving (4.3) and (4.2). Equation (4.1) is differentiable in the feasible region and (4.2)–(4.3) are strictly convex. Then again according to convergence analysis for block coordinate descent methods [2], any limit point of our CNMF/ANLS algorithm will be a stationary point.

4.2. Sparse NMF with L_1-norm constraint. The idea of imposing L_1-norm–based constraints for the purpose of achieving sparsity in the solution has been successfully utilized in a variety of problems [35]. For NMF, we propose the following formulation of NMF that imposes sparsity on the right-side factor H (SNMF/R) [16, 18],

$$\min_{W,H} \frac{1}{2} \left\{ \|A - WH\|_F^2 + \eta \|W\|_F^2 + \beta \sum_{j=1}^{n} \|\mathbf{h}_j\|_1^2 \right\} \quad \text{s.t.} \quad W, H \geq 0, \tag{4.4}$$

where $\mathbf{h}_j$ is the jth column vector of H, the parameter $\eta \geq 0$ suppresses the growth of W, and the parameter $\beta \geq 0$ balances the trade-off between the accuracy of the approximation and the sparseness of H. Note that due to the nonnegativity constraint on H, the last term in (4.4) becomes equivalent to $\beta \sum_{j=1}^{n} (\sum_{i=1}^{k} h_{ij})^2$ and accordingly (4.4) is differentiable in the feasible domain. The SNMF/R algorithm begins with the initialization of W with nonnegative values. Then it iterates the following ANLS until a convergence criterion is satisfied:

$$\min_{H\geq 0} \left\| \begin{pmatrix} W \\ \sqrt{\beta} \mathbf{e}_{1\times k} \end{pmatrix} H - \begin{pmatrix} A \\ \mathbf{0}_{1\times n} \end{pmatrix} \right\|_F^2, \tag{4.5}$$

where $\mathbf{e}_{1\times k} \in \mathbb{R}^{1\times k}$ is a row vector with all components equal to one and $\mathbf{0}_{1\times n} \in \mathbb{R}^{1\times n}$ is a zero vector, and

$$\min_{W\geq 0} \left\| \begin{pmatrix} H^T \\ \sqrt{\eta} I_k \end{pmatrix} W^T - \begin{pmatrix} A^T \\ 0_{k\times m} \end{pmatrix} \right\|_F^2, \tag{4.6}$$

where $0_{k\times m}$ is a zero matrix of size $k \times m$. Equation (4.5) minimizes the L_1-norm of each column of $H \in \mathbb{R}^{k\times n}$.

Similarly, sparsity in the NMF can be imposed on the left-side factor W (SNMF/L) through the following formulation:

$$\min_{W,H} \frac{1}{2} \left\{ \|A - WH\|_F^2 + \zeta \|H\|_F^2 + \alpha \sum_{i=1}^{m} \|\mathbf{w}_i\|_1^2 \right\} \quad \text{s.t.} \quad W, H \geq 0, \tag{4.7}$$

where $\mathbf{w}_i^T$ is the ith row vector of W, $\zeta \geq 0$ is a parameter to suppress $\|H\|_F^2$, and $\alpha \geq 0$ is a parameter to balance the trade-off between accuracy of approximation and

sparseness of W. The corresponding algorithm SNMF/L begins with an initialization of the nonnegative matrix H. Then it iterates the following ANLS until a convergence criterion is satisfied:

$$\min_{W} \left\| \begin{pmatrix} H^T \\ \sqrt{\alpha}\mathbf{e}_{1\times k} \end{pmatrix} W^T - \begin{pmatrix} A^T \\ \mathbf{0}_{1\times m} \end{pmatrix} \right\|_F^2 \quad \text{s.t.} \quad W \geq 0, \tag{4.8}$$

where $\mathbf{e}_{1\times k} \in \mathbb{R}^{1\times k}$ is a row vector whose elements are all one and $\mathbf{0}_{1\times m} \in \mathbb{R}^{1\times m}$ is a zero vector, and

$$\min_{H} \left\| \begin{pmatrix} W \\ \sqrt{\zeta} I_k \end{pmatrix} H - \begin{pmatrix} A \\ 0_{k\times n} \end{pmatrix} \right\|_F^2 \quad \text{s.t.} \quad H \geq 0, \tag{4.9}$$

where I_k is a $k \times k$ identity matrix and $0_{k\times n}$ is a zero matrix of size $k \times n$. Note that (4.8) can be rewritten as

$$\min_{W} \|H^T W^T - A^T\|_2^2 + \alpha \sum_{i=1}^{m} \left(\sum_{q=1}^{k} W^T(q,i) \right)^2 \quad \text{s.t.} \quad W \geq 0, \tag{4.10}$$

and since all elements in W are nonnegative, (4.10) in turn becomes the following by the definition of the L_1-norm of a vector:

$$\min_{W\geq 0} \left\{ \|H^T W^T - A^T\|_2^2 + \alpha \sum_{i=1}^{m} \|\mathbf{w}_i\|_1^2 \right\}, \tag{4.11}$$

which involves the L_1-norm minimization of each row of W.

An advantage of the above formulation and algorithms is that they follow the framework of the two-block coordinate descent method and therefore guarantee convergence of limit points to a stationary point. Imposing additional sparsity constraints on W or H may provide sparser factors and a simpler interpretation. However, imposing sparsity in the factors does not necessarily improve the solution or interpretation. Indeed, as the sparse constraints become stronger, the magnitude of perturbations to the basic NMF solution may become larger and the degree of simplification becomes higher.

5. Regularized NMF based on alternating nonnegativity constrained least squares. As shown in section 2, in the algorithm NMF/ANLS, one of the factors W and H is initialized and the iterations are repeated fixing one of the factors. Let us assume that H is initialized. In NMF, the columns of the computed factor W are interpreted as basis vectors, therefore, implicitly assumed to be of full rank and, in fact, many of the NMF algorithms are designed assuming that the fixed matrices H^T and W involved in the subproblems are of full rank. We propose the following regularized version of the NMF/ANLS, which we call RNMF/ANLS, where the terms $\sqrt{\alpha}I$ and $\sqrt{\beta}I$ with very small parameters $\alpha > 0$ and $\beta > 0$ are attached to the fixed matrices for the purpose of numerical stability. In RNMF/ANLS, after the matrix H is initialized the following steps are iterated: solve

$$\min_{W\geq 0} \left\| \begin{pmatrix} H^T \\ \sqrt{\alpha} I_k \end{pmatrix} W^T - \begin{pmatrix} A^T \\ 0_{k\times m} \end{pmatrix} \right\|_F^2, \tag{5.1}$$

where I_k is a $k \times k$ identity matrix and $0_{k \times m}$ is a zero matrix of size $k \times m$, and solve

$$\min_{H \geq 0} \left\| \begin{pmatrix} W \\ \sqrt{\beta} I_k \end{pmatrix} H - \begin{pmatrix} A \\ 0_{k \times n} \end{pmatrix} \right\|_F^2, \tag{5.2}$$

where $0_{k \times n}$ is a zero matrix of size $k \times n$. Similarly, one may initialize $W \in \mathbb{R}^{m \times k}$ and alternate the above in the order to solve (5.2) and then (5.1).

The above RNMF/ANLS is one way to formulate a two-block coordinate descent method for the objective function

$$\min_{W,H} \frac{1}{2} \{ \|A - WH\|_F^2 + \alpha \|W\|_F^2 + \beta \|H\|_F^2 \} \quad \text{s.t.} \quad W, H \geq 0, \tag{5.3}$$

where $\alpha \geq$ and $\beta \geq$ are very small regularization parameters. Note that the objective function (5.3) and ANLS iterations (5.1) and (5.2) are identical to the CNMF formulation and our proposed CNMF/ANLS algorithm presented in section 4.1. However, the purpose of the CNMF [30] was to obtain a sparser NMF and the role of the parameters α and β was supposed to control the sparsity of W and H. On the other hand, the purpose of the RNMF/ANLS is to impose strong convexity on the subproblems of NMF/ANLS. The role of the parameters α and β with very small values is to impose full rank on the matrices on the left side of solution matrices in the NLS subproblems. Consequently, we can guarantee that the symmetric square matrix appearing in the normal equations for solving least squares subproblems in the fast NLS algorithm [36] is symmetric positive definite with any passive set of columns, so that the solution can be computed via the Cholesky factorization.

6. Numerical experiments and discussion. In this section, we present several numerical experimental results to illustrate the behavior of our proposed algorithms and compare them to two of the most commonly used algorithms in the literature, NMF/NUR [21, 22] and NMF/ALS [1]. We implemented all algorithms in MATLAB 6.5 [26] on a P3 600MHz machine with 512MB memory.

6.1. Data sets in experiments. We have used four data sets for our empirical tests, of which two are from microarray analysis and are presented in [8, 16, 18] and the others are artificially generated. All data sets contain only nonnegative entries.

I. Data set ALLAML. The leukemia gene expression data set ALLAML [10] contains acute lymphoblastic leukemia (ALL) that has B and T cell subtypes, and acute myelogenous leukemia (AML) that occurs more commonly in adults than in children. This gene expression data set consists of 38 bone marrow samples (19 ALL-B, 8 ALL-T, and 11 AML) with 5,000 genes forming a data matrix $A \in \mathbb{R}^{5,000 \times 38}$. The gene expression values were in the range between 20 and 61,225, where a lower cutoff threshold value of 20 was used to eliminate noisy fluctuations.

II. Data set CNS. The central nervous system tumors data set CNS [33] is composed of four categories of CNS tumors with 5,597 genes. It consists of 34 samples representing four distinct morphologies: 10 classic medulloblastomas, 10 malignant gliomas, 10 rhabdoids, and 4 normals, forming a data matrix $A \in \mathbb{R}^{5,597 \times 34}$. In addition to a lower cutoff threshold value of 20, an upper cutoff threshold value of 16,000 was used to eliminate expression values that are too high and may undesirably dominate the objective function value in (1.2).

III. Artificial data sets with zero residual. We generated the first artificial data matrix A_a of size 200×50 by $A_a = W_a H_a$, where $W_a \in \mathbb{R}^{200\times 6}$ and $H_a \in \mathbb{R}^{6\times 50}$ are artificial positive matrices. The rank of A_a is 6 and a zero residual solution for the NMF with $k = 6$ exists. Accordingly, the NMF algorithms are expected to produce the solutions W and H, which give very small relative residual $\|A_a - WH\|_F/\|A_a\|_F$ with $k = 6$. We generated another artificial data matrix A_s of size $2{,}500 \times 28$ by $A_s = W_s H_s$, where $W_s \in \mathbb{R}^{2{,}500\times 3}$ and $H_s \in \mathbb{R}^{3\times 28}$ are artificial nonnegative matrices. The basis matrix W_s has columns of unit L_2-norm. The maximal value in H_s is 10^5. The rank of A_s is 3 and a zero residual solution for the NMF with $k = 3$ exists.

6.2. Convergence criteria. Reaching a smaller approximation error $\|A - W_* H_*\|_F$, where W_* and H_* are the solution matrices obtained from an algorithm for the NMF formulation in (1.2), indicates the superiority of an algorithm in terms of approximation capability. Accordingly, the convergence of the proposed algorithms may be tested by checking the decrease in the residual of the objective function $f(W, H)$. We may also test the convergence to a stationary point by checking the KKT optimality conditions. The KKT conditions shown in (2.3) can be rewritten as

$$\begin{aligned} &\min(W, \partial f(W,H)/\partial W) = 0, \\ &\min(H, \partial f(W,H)/\partial H) = 0, \end{aligned} \tag{6.1}$$

where the minimum is taken componentwise [11]. The normalized KKT residual Δ is then defined as $\Delta = \frac{\Delta_o}{\delta_W + \delta_H}$, which reflects the average of convergence errors for elements in W and H that did not converge, where

$$\begin{aligned} \Delta_o = &\sum_{i=1}^{m}\sum_{q=1}^{k} |\min(W_{iq}, (\partial f(W,H)/\partial W)_{iq})| \\ &+ \sum_{q=1}^{k}\sum_{j=1}^{n} |\min(H_{qj}, (\partial f(W,H)/\partial H)_{qj})|, \end{aligned} \tag{6.2}$$

$\delta_W = \#(\min(W, \partial f(W,H) \ / \ \partial W) \neq 0)$, and $\delta_H = \#(\min(H, \partial f(W,H) \ / \ \partial H) \neq 0)$. Then the convergence criterion is defined as

$$\Delta \leq \epsilon \Delta_1, \tag{6.3}$$

where Δ_1 is the value of Δ after one iteration and ϵ is an assigned tolerance.

6.3. Performance comparisons. In this subsection, we present performance results based on the three data sets described earlier. In the tests, we used the KKT convergence criterion shown in (6.3) with $\epsilon = 10^{-9}$.

I. Test results on the ALLAML data set. Table 6.1 shows the performance comparison among NMF/NUR, NMF/ALS, and NMF/ANLS on the ALLAML leukemia data matrix with $k = 3$. There are three clusters in this data set.[1] We report the percentage of zero elements in the computed factors W and H, relative approximation error (i.e., $\|A - WH\|_F/\|A\|_F$), the number of iterations, and computing time. The results show that to reach the same convergence criterion, NMF/NUR and NMF/ALS took much longer than NMF/ANLS, and the NMF/ALS generated the solutions with the largest relative approximation error among them. We believe

[1] The results of NMF algorithms with $k = 4$ and $k = 5$ can be found in our paper [17].

TABLE 6.1

*Performance comparison among NMF/NUR [22], NMF/ALS [1], and NMF/ANLS on the leukemia ALLAML data set with $k = 3$. We present the percentages of zero elements in W and H, relative approximation error, the number of iterations, and computing time. *For NMF/NUR, the computed W and H factors were not sparse, so the percentages of the number of nonnegative elements that are smaller than 10^{-8} in W and H are shown instead.*

Algorithms	NMF/NUR	NMF/ALS	NMF/ANLS
$\#(W = 0)$ (%)	2.71%*	2.83%	2.71%
$\#(H = 0)$ (%)	18.42%*	16.67%	18.42%
$\|A - WH\|_F / \|A\|_F$	0.5027	0.5032	0.5027
No. of iterations	5385	3670	90
Computing time	284.0 sec.	192.8 sec.	8.3 sec.

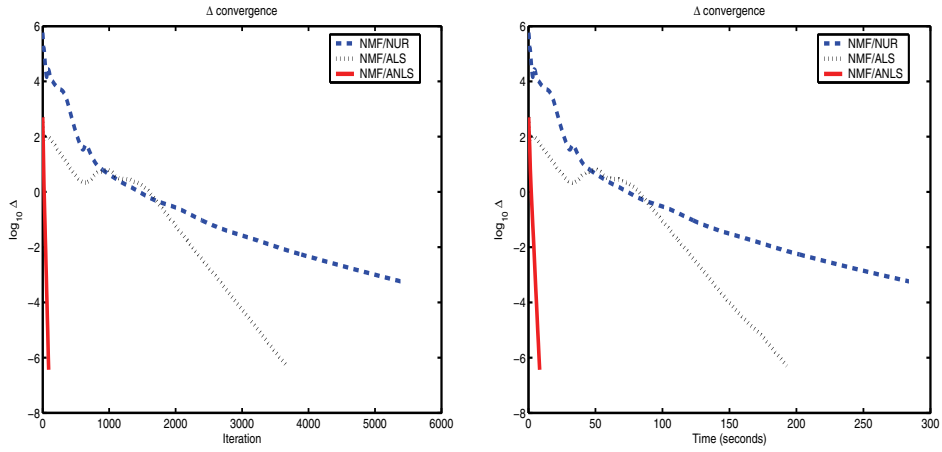


FIG. 6.1. *The values of Δ vs. the number of iterations for NMF/ANLS, NMF/NUR [22], and NMF/ALS [1] on the leukemia ALLAML data set with $k = 3$. We used the KKT convergence criterion with $\epsilon = 10^{-9}$.*

that the overall faster performance of the NMF/ANLS is a result of its convergence properties. In the factors W and H, the NMF/NUR produced very small nonnegative elements ($< 10^{-8}$) in W and H, which are not necessarily zeros, while NMF/ANLS generated the exact zero elements. This is an interesting property of the NMF algorithms and illustrates that the NMF/ANLS does better at generating sparser factors, which can be helpful in reducing computing complexity and storage requirement for handling sparse data sets.

Figure 6.1 further illustrates the convergence behavior of NMF/ANLS, NMF/NUR, and NMF/ALS on the ALLAML data set with $k = 3$. As for NMF/ALS, we solved each least squares subproblem by normal equations and set the negative values to zeros. All three algorithms began with the same random initial matrix of H_o. An additional random initial matrix of W_o was needed for NMF/NUR. The NMF/ALS generated the smallest Δ_1 (Δ value after the first iteration), whereas NMF/NUR produced the largest Δ_1. While NMF/NUR converged after more than 5,000 iterations from relatively large Δ_1, the final Δ value is still larger than those of other algorithms. We observed that the NMF/ALS algorithm required more running time than NMF/ANLS even though its subproblem (unconstrained least squares problem)

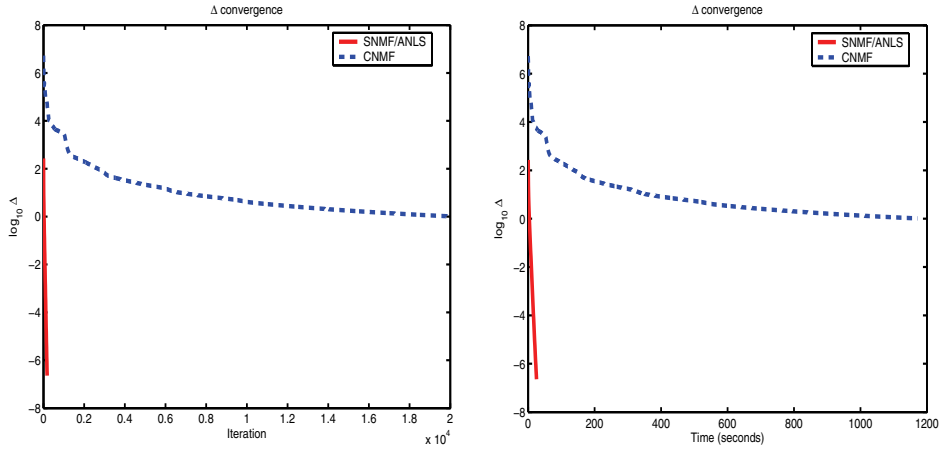


FIG. 6.2. *The values of* Δ *vs. the number of iterations for SNMF/R* [18] *with* $\beta = 0.01$ *and CNMF based on multiplicative update rules* [30] *with* $\alpha = 0$ *and* $\beta = 0.01$ *on the leukemia ALLAML data set with* $k = 3$. *We used the KKT convergence criterion with* $\epsilon = 10^{-9}$.

requires less floating point operations. This slower computational time can be ascribed to the lack of convergence property of the NMF/ALS algorithm. In this test, NMF/ANLS outperformed the others in terms of convergence speed.

Figure 6.2 illustrates the converge behavior of SNMF/R with $\beta = 0.01$ and CNMF with $\alpha = 0$ and $\beta = 0.01$ on the ALLAML data set with $k = 3$. We used the KKT convergence criterion corresponding to each of the objective functions for SNMF/R and CNMF. The η parameter in SNMF/R was set to the square of the maximal value in the ALLAML data matrix. As for CNMF, we used the CNMF algorithm based on multiplicative update rules [30] without column normalization of W in each iteration. Two algorithms began with a random initial matrix W_o that has columns of unit L_2-norm. A random initial matrix of H_o was used only in CNMF. SNMF/R generated much smaller Δ than CNMF within a short time. The percentages of zero elements in W and H obtained from SNMF/R were 2.17% and 30.70%. On the other hand, the percentages of elements in the range of $[0, 10^{-8})$ in W and H obtained from CNMF were 2.71% and 18.42% and only a small number of elements in W were exactly zeros. It illustrates that the SNMF/R is more effective in producing a sparser H.

II. Test results on the CNS tumors data set. Table 6.2 shows the performance comparison on the CNS tumors data set with various k values where NMF/ANLS was a few orders of magnitude faster than NMF/NUR. NMF/NUR did not satisfy the KKT convergence criterion within 20,000 iterations. The relative approximation errors of NMF/NUR at the last iteration were still slightly larger than those of NMF/ANLS after less than 200 iterations.

The sparsity obtained from NMF/NUR or NMF/ANLS is a general result due to nonnegativity constraints. Even when the original data set has no zero element, the factors W and H may have zero components. In case of NMF/ANLS, this becomes clear when we note that at the core of NMF/ANLS is the active set–based iterations, and in each iteration the solution components that correspond to the active set index are set to be zeros.

TABLE 6.2

*Performance comparison between NMF/NUR [22] and NMF/ANLS on the CNS tumors data set. We report the percentages of zero elements in W and H, relative approximation error, the number of iterations, and computing time (in seconds). *For NMF/NUR, the computed W and H factors were not sparse, so the percentages of the number of nonnegative elements that are smaller than 10^{-8} in W and H are shown instead.*

Algorithm	NMF/NUR		
Reduced rank k	3	4	5
$\#(\lvert W_{ij}\rvert < 10^{-8})$ (%)	8.70%*	9.05%*	12.32%*
$\#(\lvert H_{ij}\rvert < 10^{-8})$ (%)	18.63%*	25.00%*	25.29%*
$\lVert A - WH\rVert_F / \lVert A\rVert_F$	0.40246175083	0.37312046970	0.35409585961
No. of iterations	20000	20000	20000
Computing time	1310.0 sec.	1523.0 sec.	1913.9 sec.
Algorithm	NMF/ANLS		
Reduced rank k	3	4	5
$\#(W = 0)$ (%)	8.69%	9.03%	12.07%
$\#(H = 0)$ (%)	18.63%	25.00%	27.06%
$\lVert A - WH\rVert_F / \lVert A\rVert_F$	0.40246175028	0.37312046948	0.35409574992
No. of iterations	150	130	130
Computing time	14.8 sec.	16.6 sec.	20.4 sec.

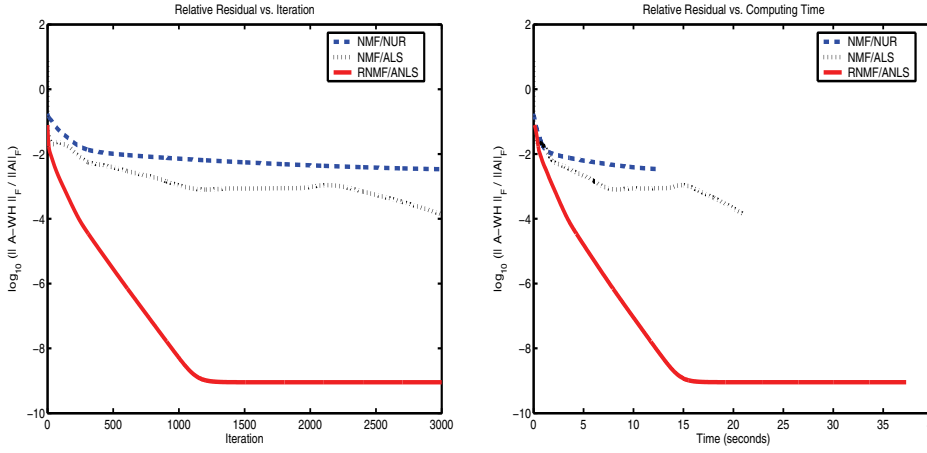


FIG. 6.3. *The relative residuals vs. the number of iterations for RNMF/ANLS with $\alpha = \beta = 10^{-8}$, NMF/NUR [22], and NMF/ALS [1] with $k = 6$ for 3,000 iterations on the first artificial data matrix $A_a = W_a H_a$ of size 200×50, where $W_a \in \mathbb{R}^{200\times 6}$ and $H_a \in \mathbb{R}^{6\times 50}$ are artificial positive matrices, and $\mathrm{rank}(A_a) = 6$.*

III. Test results on the zero residual artificial data sets. Figure 6.3 shows the performance of the three NMF algorithms, RNMF/ANLS, NMF/NUR, and NMF/ALS, on the first artificial data matrix $A_a = W_a H_a$ of size 200×50 where $W_a \in \mathbb{R}^{200\times 6}$ and $H_a \in \mathbb{R}^{6\times 50}$ are artificial positive matrices. The relative residuals versus iteration or computing time are shown. We used $\alpha = \beta = 10^{-8}$ for the RNMF/ANLS and implemented NMF/ALS with pseudoinverse. We note that NMF/ALS sometimes generated ill-conditioned W and H when negative values are set to zeros, which may happen even when we solve the least squares problem by a

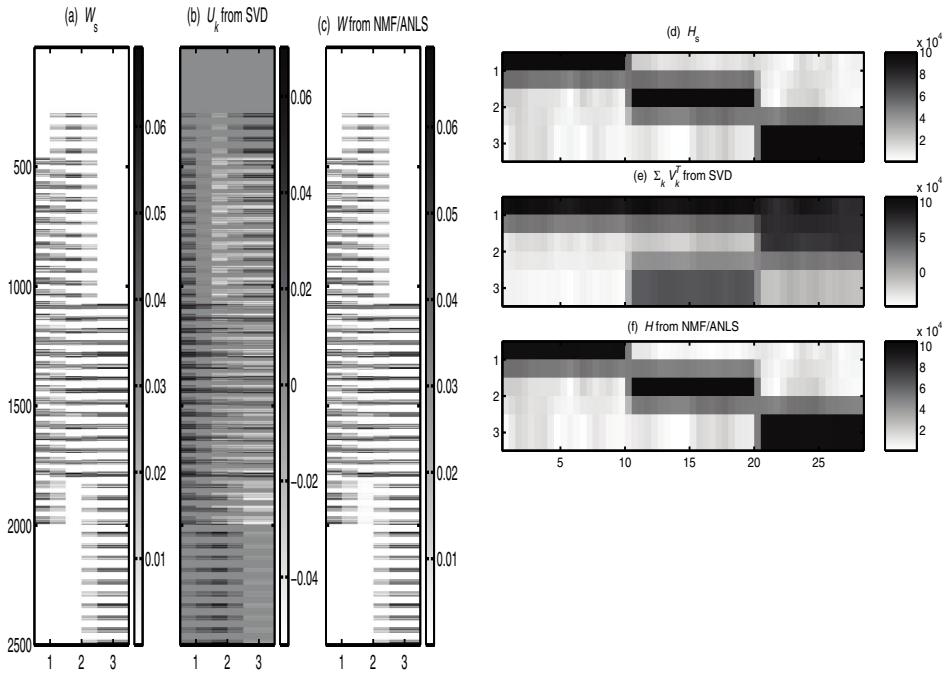


FIG. 6.4. *The factors obtained from the truncated SVD* [9] *($A_s \approx U_k \Sigma_k V_k^T$) and NMF/ANLS ($A_s \approx WH$ s.t. $W, H \geq 0$) with $k = 3$ on the second artificial data matrix $A_s = W_s H_s$ of size* $2{,}500 \times 28$*, where $W_s \in \mathbb{R}^{2,500 \times 3}$ and $H_s \in \mathbb{R}^{3 \times 28}$ are artificial nonnegative matrices, and* $\mathrm{rank}(A_s) = 3$. *The gray scale indicates the values of the elements in the matrix.*

stable algorithm. In the worst case, the entire row or column in the matrices W or H may become zero. The RNMF/ANLS rapidly converged, while NMF/NUR did not converge to near zero residual within 3,000 iterations. The relative residual in the middle of iterative optimization of NMF/ALS sometimes increased.

Figure 6.4 shows the comparison between the truncated SVD [9] and NMF/ANLS on the second artificial data matrix $A_s = W_s H_s$ of size $2{,}500 \times 28$, where $W_s \in \mathbb{R}^{2,500 \times 3}$ and $H_s \in \mathbb{R}^{3 \times 28}$ are artificial nonnegative matrices. We presented U_k and $\Sigma_k V_k^T$ obtained from the truncated SVD ($A_s \approx U_k \Sigma_k V_k^T$) with $k = 3$. We also illustrated W and H obtained from NMF/ANLS ($A_s \approx WH$ s.t. $W, H \geq 0$) with $k = 3$. Although the approximation error of NMF/ANLS was larger than that of the truncated SVD, it surprisingly recovered W_s and H_s factors much better. Our NMF algorithm can be utilized for blind source separation when basis vectors are nonnegative and observations are nonsubtractive combinations of basis vectors.

6.4. Summary of experimental results. In our tests, the convergence of NMF/NUR was slower and, due to this, the algorithm was often prematurely terminated before it reached a convergence criterion, whether it was based on the relative residual or the KKT residual. The NMF/ALS does not provide a solution in a least squares sense for each nonnegativity constrained subproblem although the problem is formulated as a least squares problem. Therefore, its convergence is difficult to analyze and exhibits nonmonotonic changes in the objective function value throughout the iterations. On the other hand, NMF/ANLS generated solutions with satisfactory accuracy within a reasonable time.

An algorithm for nonnegativity constrained least squares is an essential component of NMF/ANLS. There are several ways to solve the NLS problem with multiple right-hand sides and we chose van Benthem and Keenan's NLS algorithm [36]. This algorithm is based on the active set method that is guaranteed to terminate in [illegible] of matrix computations. Some other implementations of NLS are based on traditional gradient descent or quasi-Newton optimization methods. They are iterative methods that require explicit convergence check parameters. Their speed and accuracy depend on their convergence check parameters.

7. Summary and discussion. We have introduced the NMF algorithms based on alternating nonnegativity constrained least squares, for which every limit point is a stationary point. The core of our algorithm is the nonnegativity constrained least squares algorithm for multiple right-hand sides based on the active set method, which terminates in [illegible]. We applied the well-known convergence theory for block coordinate descent methods in bound constrained optimization and built a rigorous convergence criterion based on the KKT conditions.

We have established a framework of NMF/ANLS which is theoretically sound and practically efficient. This framework was utilized to design formulations and algorithms for sparse NMFs and regularized NMF. Some theoretical characteristics of our proposed algorithms explain their superior behavior shown in the test results. The NMF algorithms based on gradient descent method exhibit slow convergence. Thus, it is possible, though undesirable, to use premature solutions for data analysis owing to termination before convergence, which may sometimes lead to unreliable conclusions. The [illegible] NMF/ALS algorithm [1] sets the negative components in the unconstrained least squares solution to zero. Although the inexact method may solve the subproblems faster, its convergence behavior is problematic. On the other hand, our algorithm satisfies the nonnegativity constraints exactly in each subproblem and shows faster overall convergence. The converged solutions obtained from our algorithms make it possible to reach more physically reliable conclusions in many applications of NMF. The NMF/ANLS can be applied to a wide variety of practical problems in the fields of text data mining, image analysis, bioinformatics, computational biology, and so forth, especially when preserving nonnegativity is beneficial to meaningful interpretation.

Acknowledgment. We would like to thank Prof. Chih-Jen Lin and Prof. Luigi Grippo for discussions on the convergence properties.

REFERENCES

[1] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, *Algorithms and applications for approximate nonnegative matrix factorization*, Comput. Statist. Data Anal., 52 (2007), pp. 155–173.

[2] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[3] R. Bro and S. de Jong, *A fast non-negativity-constrained least squares algorithm*, J. Chemometrics, 11 (1997), pp. 393–401.

[4] J. P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, *Metagenes and molecular pattern discovery using matrix factorization*, Proc. Natl. Acad. Sci. USA, 101 (2004), pp. 4164–4169.

[5] M. Chu, F. Diele, R. Plemmons, and S. Ragni, *Optimality, Computation and Interpretation of Nonnegative Matrix Factorization*, preprint, 2004.

[6] C. Ding, T. Li, W. Peng, and H. Park, *Orthogonal nonnegative matrix tri-factorizations for clustering*, in Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD 2006), ACM Press, New York, 2006, pp. 126–135.

[7] D. Dueck, Q. D. Morris, and B. J. Frey, *Multi-way clustering of microarray data using probabilistic sparse matrix factorization*, Bioinform., 21 (2005), pp. i144–i151.
[8] Y. Gao and G. Church, *Improving molecular cancer class discovery through sparse non-negative matrix factorization*, Bioinform., 21 (2005), pp. 3970–3975.
[9] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[10] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, *Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring*, Science, 286 (1999), pp. 531–537.
[11] E. F. Gonzales and Y. Zhang, *Accelerating the Lee-Seung Algorithm for Non-negative Matrix Factorization*, Tech. report, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2005.
[12] L. Grippo and M. Sciandrone, *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints*, Oper. Res. Lett., 26 (2000), pp. 127–136.
[13] P. O. Hoyer, *Non-negative sparse coding*, in Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, 2002, pp. 557–565.
[14] P. O. Hoyer, *Non-negative matrix factorization with sparseness constraints*, J. Mach. Learn. Res., 5 (2004), pp. 1457–1469.
[15] D. Kim, S. Sra, and I. S. Dhillon, *Fast Newton-type methods for the least squares nonnegative matrix approximation problem*, in Proceedings of the Seventh SIAM International Conference on Data Mining, SIAM, Philadelphia, 2007, pp. 343–354.
[16] H. Kim and H. Park, *Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares*, in Proceedings of the IASTED International Conference on Computational and Systems Biology (CASB2006), D.-Z. Du, ed., 2006, pp. 95–100.
[17] H. Kim and H. Park, *Cancer class discovery using non-negative matrix factorization based on alternating non-negativity-constrained least squares*, in Proceedings of the Third International Symposium on Bioinformatics Research and Applications, ISBRA 2007, Lecture Notes in Comput. Sci. 4463, Springer-Verlag, New York, 2007, pp. 477–487.
[18] H. Kim and H. Park, *Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis*, Bioinform., 23 (2007), pp. 1495–1502.
[19] P. M. Kim and B. Tidor, *Subsystem identification through dimensionality reduction of large-scale gene expression data*, Genome Res., 13 (2003), pp. 1706–1718.
[20] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice–Hall, Englewood Cliffs, NJ, 1974.
[21] D. D. Lee and H. S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.
[22] D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization*, in Advances in Neural Information Processing 13, MIT Press, Cambridge, MA, 2001, pp. 556–562.
[23] S. Z. Li, X. Hou, H. Zhang, and Q. Cheng, *Learning spatially localized parts-based representations*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2001, pp. 207–212.
[24] C. J. Lin, *Projected Gradient Methods for Non-negative Matrix Factorization*, Tech. Report Information and Support Service ISSTECH-95-013, Department of Computer Science, National Taiwan University, 2005.
[25] W. Liu and J. Yi, *Existing and New Algorithms for Nonnegative Matrix Factorization*, Tech. report, University of Texas, Austin, 2003.
[26] *MATLAB User's Guide*, The MathWorks, Natick, MA, 1992.
[27] P. Paatero and U. Tapper, *Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics, 5 (1994), pp. 111–126.
[28] H. Park and H. Kim, *One-sided non-negative matrix factorization and non-negative centroid dimension reduction for text classification*, in Proceedings of the Workshop on Text Mining at the Sixth SIAM International Conference on Data Mining, M. Castellanos and M. W. Berry, eds., SIAM, Philadelphia, 2006.
[29] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, *Nonsmooth nonnegative matrix factorization (nsNMF)*, IEEE Trans. Pattern Anal. Machine Intell., 28 (2006), pp. 403–415.
[30] V. P. Pauca, J. Piper, and R. J. Plemmons, *Nonnegative matrix factorization for spectral data analysis*, Linear Algebra Appl., 416 (2006), pp. 29–47.
[31] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, *Text mining using non-negative matrix factorizations*, in Proceedings of the Fourth SIAM International Conference on Data Mining, SIAM, Philadelphia, 2004, pp. 452–456.

[32] J. Piper, V. P. Pauca, R. J. Plemmons, and M. Giffin, *Object characterization from spectral data using nonnegative factorization and information theory*, in Proceedings of the Amos Technical Conference, 2004.

[33] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub, *Prediction of central nervous system embryonal tumour outcome based on gene expression*, Nature, 415 (2002), pp. 436–442.

[34] S. Sra and I. S. Dhillon, *Nonnegative Matrix Approximation: Algorithms and Applications*, Tech. Report 06-27, University of Texas, Austin, 2006.

[35] R. Tibshirani, *Regression shrinkage and selection via LASSO*, J. Roy. Statist. Soc. B, 58 (1996), pp. 267–288.

[36] M. H. van Benthem and M. R. Keenan, *Fast algorithm for the solution of large-scale nonnegativity-constrained least squares problems*, J. Chemometrics, 18 (2004), pp. 441–450.

[37] R. Zdunek and A. Cichocki, *Non-negative matrix factorization with quasi-Newton optimization*, in Proceedings of the Eighth International Conference on Artificial Intelligence and Soft Computing (ICAISC), 2006, pp. 870–879.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 731–740

ON THE MINIMUM RANK AMONG POSITIVE SEMIDEFINITE MATRICES WITH A GIVEN GRAPH*

MATTHEW BOOTH†, PHILIP HACKNEY‡, BENJAMIN HARRIS§, CHARLES R. JOHNSON¶, MARGARET LAY‖, LON H. MITCHELL#, SIVARAM K. NARAYAN††, AMANDA PASCOE‡‡, KELLY STEINMETZ§§, BRIAN D. SUTTON¶¶, AND WENDY WANG‖‖

Abstract. Let $\mathcal{P}(G)$ be the set of all positive semidefinite matrices whose graph is G, and $\mathrm{msr}(G)$ be the minimum rank of all matrices in $\mathcal{P}(G)$. Upper and lower bounds for $\mathrm{msr}(G)$ are given and used to determine $\mathrm{msr}(G)$ for some well-known graphs, including chordal graphs, and for all simple graphs on less than seven vertices.

Key words. rank, positive semidefinite, graph of a matrix

AMS subject classifications. 15A18, 15A57, 05C50

DOI. 10.1137/050629793

1. Introduction. If A is an n-by-n Hermitian matrix, then its graph $G(A)$ is the undirected, simple graph on vertices $\{1, 2, \ldots, n\}$, which has an edge between vertices i and j if and only if the i, j entry of A is nonzero and $i \neq j$. The graph is independent of the real diagonal entries of A. The set of all Hermitian matrices that share a common graph G is denoted $\mathcal{H}(G)$: $\mathcal{H}(G) = \{ A \mid A = A^*, G(A) = G \}$. If G is a simple connected graph, then matrices in $\mathcal{H}(G)$ may be viewed as the discrete version of the continuous Schrödinger operators with magnetic fields [3].

The possible multiplicities of the eigenvalues among matrices in $\mathcal{H}(G)$ have been of much recent interest [6, 7, 9, 10, 11, 13]. It is known, for example, that if G is a tree, then the smallest eigenvalue of any matrix in $\mathcal{H}(G)$ has multiplicity one [10, Corollary

*Received by the editors April 22, 2005; accepted for publication (in revised form) by P. Benner February 8, 2008; published electronically July 3, 2008.
http://www.siam.org/journals/simax/30-2/62979.html
†Department of Mathematics, Oberlin College, Oberlin, OH 44074 (Matthew.Booth@oberlin.edu). This author's research was supported by NSF grant DMS 99-87803.
‡Department of Mathematics, Purdue University, West Lafayette, IN 47907-2067 (phackney@math.purdue.edu).
§Department of Mathematics, Brown University, Providence, RI 02912 (Benjamin_Harris@brown.edu). This author's research was supported by NSF grant DMS 02-43674.
¶Department of Mathematics, College of William and Mary, Williamsburg, VA 23187-8795 (crjohnso@math.wm.edu). This author's research was supported by NSF grant DMS 99-87803.
‖Department of Mathematics and Computer Science, Grinnell College, Grinnell, IA 50112-1690 (laymarga@grinnell.edu). This author's research was supported by NSF grant DMS 02-43674.
#Department of Mathematics, Virginia Commonwealth University, Richmond, VA 23284-2014 (lmitchell2@vcu.edu).
††Corresponding author. Department of Mathematics, Central Michigan University, Mount Pleasant, MI 48859 (sivaram.narayan@cmich.edu). This author's research was supported by NSF grant DMS 02-43674.
‡‡Department of Mathematics, Furman University, Greenville, SC 29613-1148 (amanda.pascoe@furman.edu). This author's research was supported by NSF grant DMS 02-43674.
§§Department of Mathematics, Indiana University, Bloomington, IN 47405 (kellyjs82@yahoo.com). This author's research was supported by NSF grant DMS 02-43674.
¶¶Department of Mathematics, Randolph–Macon College, Ashland, VA 23005 (bsutton@rmc.edu). This author's research was supported by NSF grant DMS 99-87803.
‖‖Department of Mathematics, Duke University, Durham, NC 27708-0320 (wendy.wang@duke.edu). This author's research was supported by NSF grant DMS 02-43674.

3.9]. This implies that any Hermitian positive semidefinite (psd) matrix whose graph is a tree has rank at least $n-1$. The Laplacian matrix of a tree on n vertices is a psd matrix with rank equal to $n-1$ [14]. A converse to this statement, that, for any nontree the minimum rank of a psd matrix is less than $n-1$, was proved independently (Lemma 5, [8] and Theorem 4.1, [17]). This raises the following interesting question, given a graph G, what is the minimum rank among psd matrices in $\mathcal{H}(G)$?

Let $\mathcal{P}(G)$ denote the psd matrices in $\mathcal{H}(G)$. Define the [illegible] G, $\operatorname{msr}(G)$, as $\min\{\operatorname{rank} A : A \in \mathcal{P}(G)\}$. We present here some results about $\operatorname{msr}(G)$, which give $\operatorname{msr}(G)$ for every chordal graph and for most graphs on fewer than seven vertices. The few exceptions can be handled by separate arguments. It is equally interesting to find the minimum psd rank over the symmetric real matrices instead of Hermitian matrices. It is not known if these two problems are different, though there can be differences in some related problems [12].

If G is not connected, it is clear that $\operatorname{msr}(G)$ is the sum of the minimum semidefinite ranks of each of G's connected components, so that we may (and do) confine our attention to connected graphs. Note that, if G is a connected graph with two or more vertices, the diagonal entries of $A \in \mathcal{P}(G)$ are positive.

2. Lower bounds using induced subgraphs. We will obtain several lower bounds using induced subgraphs. An [illegible] H of a graph G is obtained by deleting all vertices except for the vertices in a subset S. Since a principal submatrix of a psd matrix is psd [5, p. 397], and the rank of a submatrix can never be greater than that of the matrix, we have the following.

LEMMA 2.1. [illegible] $\operatorname{msr}(H) \leq \operatorname{msr}(G)$

Equality can occur in the inequality of Lemma 2.1 in important ways; of course, strict inequality is common. One case of equality is that in which the induced subgraph is the result of the deletion of a duplicate vertex from G. For a vertex w, let $n(w)$ denote the set of all vertices adjacent to w. The [illegible] of w is $n(w) \cup \{w\}$. A vertex u is a [illegible] of a vertex v of G if u and v are adjacent, and their closed neighborhoods are the same. We denote the induced subgraph of G resulting from the deletion of a vertex u by $G - u$. We then have the following.

PROPOSITION 2.2. [illegible] $\operatorname{msr}(G-u) = \operatorname{msr}(G)$

[illegible]. From Lemma 2.1, $\operatorname{msr}(G-u) \leq \operatorname{msr}(G)$. Let $A' \in \mathcal{P}(G-u)$ be a psd matrix such that $\operatorname{rank} A' = \operatorname{msr}(G-u)$. By permuting the rows and columns of A' let the first row and column of A' correspond to the vertex v. If $A' = B^*B$, then consider $A = \begin{bmatrix} B^* \\ e_1^{\mathrm{T}} B^* \end{bmatrix} \begin{bmatrix} B & Be_1 \end{bmatrix}$ where $e_1^{\mathrm{T}} = (1, 0, \ldots, 0)$. Then $\operatorname{rank} A = \operatorname{rank} A'$ and $A \in \mathcal{P}(G)$. Thus $\operatorname{msr}(G) \leq \operatorname{msr}(G-u)$. □

From a sequential deletion of duplicate vertices and application of Proposition 2.2 we get the following.

COROLLARY 2.3. [illegible] $\operatorname{msr}(H) = \operatorname{msr}(G)$

[illegible] 2.4. As an easy consequence of Corollary 2.3, we obtain that, for $n \geq 2$, $\operatorname{msr}(K_n) = 1$ where K_n denotes the complete graph on n vertices. Note that Proposition 2.2 is incorrect if applied to two [illegible] vertices with the same neighbors. To see this, let G be K_4 minus an edge. Deletion of a degree 3 vertex gives $\operatorname{msr}(G) = 2$ using Proposition 2.2, but deletion of a degree 2 vertex results in K_3 whose msr equals one.

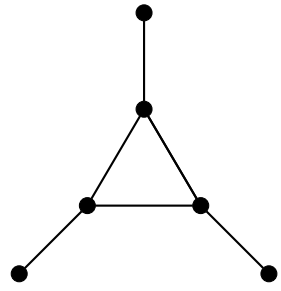


FIG. 2.1. $\mathrm{fm}(G) = 4$.

Another important application of Lemma 2.1 is that in which H is an induced tree on the maximum possible number of vertices as we know the msr for any tree. For a graph G, we consider its "tree size," denoted $\mathrm{ts}(G)$, which is the number of vertices in a maximum induced tree [4]. As already noted, when T is a tree, $\mathrm{msr}(T)$ is one less than the number of vertices of T. This fact, combined with Lemma 2.1, immediately gives the following.

LEMMA 2.5. [illegible] $\mathrm{msr}(G) \geq \mathrm{ts}(G) - 1$

As mentioned in the introduction, equality in Lemma 2.5 occurs whenever G is a tree. It also occurs for any nontree G on n vertices for which $\mathrm{ts}(G) = n - 1$; in this case $\mathrm{msr}(G) \geq n - 2$ by Lemma 2.5, and $\mathrm{msr}(G) \leq n - 2$ because G is not a tree. Thus $\mathrm{msr}(G) = n - 2$. For example, if G is a cycle on n vertices, the tree size is $n - 1$ (because deletion of any one vertex leaves a path on $n - 1$ vertices). Therefore, the msr of a cycle on n vertices is $n - 2$ (cf. [17, Theorem 4.3]).

For an induced forest of G with components $T_1, T_2, \ldots, T_k$, count $\mathrm{ts}(T_1) + \mathrm{ts}(T_2) + \cdots + \mathrm{ts}(T_k) -$ (the number of components that are not isolated vertices). Among all the induced forests of G maximize this count and call this result $\mathrm{fm}(G)$, the "forest measure" of G. Any isolated vertices occurring in an induced subgraph of a connected graph G contribute 1, rather than 0, to $\mathrm{msr}(G)$, as an irreducible psd matrix has positive diagonal entries. We then have the following.

PROPOSITION 2.6. [illegible] $\mathrm{msr}(G) \geq \mathrm{fm}(G) \geq \mathrm{ts}(G) - 1$

Figure 2.1 illustrates that strict inequality is possible in the second inequality of Proposition 2.6, as $\mathrm{fm}(G) = 4$ by deleting any single interior vertex.

One special case of an induced forest is an induced set of isolated vertices. The maximum cardinality of such a set is the [illegible] $i(G)$, the greatest number of vertices among which there are no edges. Clearly $\mathrm{fm}(G) \geq i(G)$, so that we have the following.

COROLLARY 2.7. [illegible] G, $\mathrm{msr}(G) \geq i(G)$

Suppose G is a connected graph with vertex set $V = \{v_1, v_2, \ldots, v_n\}$. We call a set of vectors $\overrightarrow{V} = \{\overrightarrow{v_1}, \overrightarrow{v_2}, \ldots, \overrightarrow{v_n}\}$ in $\mathbb{C}^m$ a [illegible] (or [illegible]) of G if

$$\begin{bmatrix} \overrightarrow{v_1} \\ \overrightarrow{v_2} \\ \vdots \\ \overrightarrow{v_n} \end{bmatrix} \begin{bmatrix} \overrightarrow{v_1} \\ \overrightarrow{v_2} \\ \vdots \\ \overrightarrow{v_n} \end{bmatrix}^* = A \in \mathcal{P}(G).$$

In other words, we associate a vector $\overrightarrow{v_i} \in \mathbb{C}^m$ to each vertex $v_i \in V(G)$ such that, for $i \neq j$, $\langle \overrightarrow{v_i}, \overrightarrow{v_j} \rangle \neq 0$ if v_i and v_j are adjacent in G, and $\langle \overrightarrow{v_i}, \overrightarrow{v_j} \rangle = 0$ if v_i and v_j

are not adjacent. Since every psd matrix $A \in \mathcal{P}(G)$ can be written as $A = B^*B$ for some matrix B, we can always find a vector representation of $G(A)$ that produces A. Also, the rank of the matrix and the dimension of the span of the vectors in the vector representation (which we call the [illegible] of the vector representation) are always the same [5, p. 408].

We end this section by giving a sufficient condition on G so that $\operatorname{msr}(G) = \operatorname{ts}(G) - 1$. To prove the result we need the following lemma.

LEMMA 2.8. [illegible] $X_1, \ldots, X_m$, $X_i \subseteq \mathbb{C}^n$, $1 \le i \le m$, [illegible] $G_1, \ldots, G_m$ [illegible] G [illegible]

- [illegible] v, w [illegible] G [illegible] i [illegible] v [illegible] w [illegible] G_i [illegible]
- [illegible] v, w [illegible] G [illegible] $\overrightarrow{x_v}$ [illegible] v, [illegible] X_i [illegible] $\overrightarrow{x_w}$ [illegible] w, [illegible] X_j, $\langle \overrightarrow{x_v}, \overrightarrow{x_w} \rangle = 0$

[illegible] X [illegible] G [illegible]

$$\operatorname{rank} X \le \operatorname{rank}\left(\bigcup_{1 \le i \le m} \operatorname{span} X_i \right) \le \sum_{1 \le i \le m} \operatorname{rank} X_i.$$

[illegible]. We prove the statement for the case of two vector representations as the result can be easily generalized. Let $X_1 = \{\overrightarrow{x_i}\}$ and $X_2 = \{\overrightarrow{w_i}\}$ be vector representations of subgraphs G_1 and G_2 of a graph G. Extend X_1 and X_2 to represent all of the vertices of G by adding copies of the zero vector if need be. We claim there exists $c \in \mathbb{R}$ such that $\{\overrightarrow{x_i} + c\overrightarrow{w_i}\}$ is a vector representation of G.

If $(v_i, v_j) \notin E$, then $\langle \overrightarrow{x_i}, \overrightarrow{x_j} \rangle = \langle \overrightarrow{w_i}, \overrightarrow{w_j} \rangle = \langle \overrightarrow{x_i}, \overrightarrow{w_j} \rangle = \langle \overrightarrow{w_i}, \overrightarrow{x_j} \rangle = 0$. This implies that $\langle \overrightarrow{x_i} + c\overrightarrow{w_i}, \overrightarrow{x_j} + c\overrightarrow{w_j} \rangle = 0$ for any $c \in \mathbb{C}$. If v_i and v_j are adjacent, then $\{\langle \overrightarrow{x_i} + c\overrightarrow{w_i}, \overrightarrow{x_j} + c\overrightarrow{w_j} \rangle\}$ is a set of quadratics in c having finitely many roots. Thus we may choose $c \in \mathbb{R}$ so that $\{\overrightarrow{x_i} + c\overrightarrow{w_i}\}$ is a vector representation of G. □

Suppose T is a maximum induced tree. If w is a vertex not belonging to T, denote by $\mathcal{E}(w)$ the edge set of all paths in T between every pair of the vertices of T that are adjacent to w.

THEOREM 2.9. [illegible] G, $\operatorname{msr}(G) = \operatorname{ts}(G) - 1$ [illegible] ⊛ [illegible]

⊛ [illegible] T [illegible] u [illegible] w [illegible] T, $\mathcal{E}(u) \cap \mathcal{E}(w) \neq \varnothing$ [illegible] u [illegible] w [illegible] G

[illegible]. If G is a tree, we have already seen that $\operatorname{msr}(G) = \operatorname{ts}(G) - 1$. If G is not a tree, we will cover G with subgraphs that have vector representations satisfying the conditions of Lemma 2.8. If ⊛ holds for a maximum induced tree T of G, then every vertex w not on T must be adjacent to some vertex on T. Moreover, by the definition of T, w is adjacent to at least two vertices of T. Assign an orthonormal set of vectors $\{\overrightarrow{x_e}\}$ of dimension $(\operatorname{ts}(G) - 1)$ to the edges of T, one vector per edge. If $v \in V(T)$, assign the vector $\overrightarrow{v} = \sum_e \overrightarrow{x_e}$ to v, where the summation is over all edges incident to v. This gives a vector representation $\overrightarrow{T}$ of T.

For any path $p = (e_1, e_2, \ldots, e_m)$ in T, let

$$\overrightarrow{p} = \sum_{j=1}^{m} (-1)^j \overrightarrow{x_{e_j}}.$$

Given a vertex w not on T and an adjacent vertex v_1 on T, w must have another neighbor v_2 on T. If p is a path between v_1 and v_2 in T, letting $\overrightarrow{p}$ represent w and $\overrightarrow{T}$ represent T yields a vector representation of a subgraph of G containing the edge between w and v_1.

Given two vertices w_1 and w_2 not on T that are adjacent, by $\circledast$ there exist intersecting paths p_1 and p_2 in T so that the end vertices of p_i are neighbors of w_i, $i = 1, 2$. Letting $\overrightarrow{p_i}$ represent w_i for $i = 1, 2$ and $\overrightarrow{T}$ represent T yields a representation of a subgraph of G containing the edge connecting w_1 and w_2.

By construction, these representations cover all the edges of G and are contained in $\operatorname{span}\{\overrightarrow{x_e}\colon e \text{ an edge of } T\}$. We now show that these representations satisfy the conditions of Lemma 2.8. If v and w are adjacent in G, we have explicitly constructed above a representation of a subgraph of G in which v and w are adjacent.

If v and w are not adjacent, there are three cases to consider. First, if v and w are both vertices in T, then in any two representations, v and w are represented by the corresponding vectors in $\overrightarrow{T}$, which are orthogonal. For other cases, first notice that, if a vertex w is not on T, then w is represented by $\overrightarrow{p}$ derived from a path p. If v is a vertex on T not adjacent to w, then v cannot be an endpoint of p. Thus the vector representing v in $\overrightarrow{T}$ is orthogonal to $\overrightarrow{p}$. Suppose v and w are both not on T and are not adjacent in G. The vectors $\overrightarrow{q}$ and $\overrightarrow{p}$ representing v and w, respectively, are derived from paths q and p, respectively. By $\circledast$ the paths p and q have no edges in common and thus $\overrightarrow{p}$ and $\overrightarrow{q}$ must be orthogonal. Applying Lemma 2.8 we get $\operatorname{msr}(G) \leq \operatorname{ts}(G) - 1$. □

3. Chordal graphs. The sum of two psd matrices is psd, and the rank of a sum is never more than the sum of the ranks [5, p. 13]. If we cover all of the edges of a graph G with (not necessarily induced) subgraphs of known msr, this can lead to useful upper bounds for $\operatorname{msr}(G)$. First, suppose that G is labeled and that $G_1, \ldots, G_k$ are (labeled) subgraphs of G, that is, each $G_i, i = 1, \ldots, k$ is the result of deleting some edges and/or vertices from G. We say that $G_1, \ldots, G_k$ [illegible] G if each vertex of G is a vertex of at least one G_i, and for every pair of adjacent vertices v, w of G, v and w are adjacent in at least one G_i. The cover $C_1, \ldots, C_k$ of G is called a [illegible] of G if each of $C_1, \ldots, C_k$ is a clique of G. The [illegible] $\operatorname{cc}(G)$ (see [15]) of G is the minimum value of k for which there is a clique cover $C_1, \ldots, C_k$ of G.

PROPOSITION 3.1. [illegible] G, $\operatorname{msr}(G) \leq \operatorname{cc}(G)$ [illegible]. The proof follows from Lemma 2.8 and Remark 2.4. □

Since the clique cover number of a cycle on $n \geq 4$ vertices is n but its msr is $n-2$, strict inequality is possible in Proposition 3.1.

Given a vector representation $\overrightarrow{V}$ of G, with $\overrightarrow{v}$ representing vertex v, replace each vector $\overrightarrow{w} \in \overrightarrow{V}$ with the orthogonal projection

$$\overrightarrow{w} - \frac{\langle \overrightarrow{v}, \overrightarrow{w} \rangle}{\langle \overrightarrow{v}, \overrightarrow{v} \rangle} \overrightarrow{v}$$

to yield a set of vectors denoted $\overrightarrow{V} \ominus \overrightarrow{v}$. It is easily verified that $\operatorname{rank}(\overrightarrow{V})$ is one more than $\operatorname{rank}(\overrightarrow{V} \ominus \overrightarrow{v})$.

Consider the graph corresponding to $\overrightarrow{V} \ominus \overrightarrow{v}$. It is obtained from the original graph G, first by removing the vertex v and then modifying the graph in the following manner: For $u, w \in n(v)$, if (u, w) is not an edge of G, then (u, w) is an edge of the modified graph and if (u, w) is an edge of G, then (u, w) may or may not be an edge

of the modified graph. Notice in the latter case that the "may or may not" depends on the choice of vector representation $\overrightarrow{V}$. In what follows, we consider graphs which have multiple edges. This allows us to define below a graph $G \ominus v$, which better captures the relationship between $\overrightarrow{V} \ominus \overrightarrow{v}$ and the "orthogonal removal of vertex v."

Following van der Holst [17], let G be an undirected graph with no loops but possibly multiple edges, with vertex set $V = \{1, 2, \ldots, n\}$. Let $\mathcal{H}_G$ be the set of all n-by-n Hermitian matrices $A = [a_{ij}]$ such that

- $a_{ij} \neq 0$ if i and j are connected by exactly one edge;
- $a_{ij} = 0$ if i and j are not adjacent, and $i \neq j$.

Notice that we make no restriction on a_{ij} if i and j are connected by more than one edge. Now, $\overrightarrow{V} = \{\overrightarrow{v_1}, \ldots, \overrightarrow{v_n}\}$ in $\mathbb{C}^m$ is a [illegible] of a graph G with multiple edges when $\langle \overrightarrow{v_i}, \overrightarrow{v_j} \rangle \neq 0$ if i and j are connected by a single edge and $\langle \overrightarrow{v_i}, \overrightarrow{v_j} \rangle = 0$ if i and j are not adjacent.

Let G be a graph (with multiple edges). The graph $G \ominus v$, called the [illegible] v [illegible] G, is obtained as follows: In the induced subgraph $G - v$ of G, between any $u, w \in n(v)$ add $e - 1$ edges, where e is the sum of the number of edges between u and v and the number of edges between w and v.

[illegible] 3.2. If $\overrightarrow{V}$ is a vector representation of a graph G, then $\overrightarrow{V} \ominus \overrightarrow{v}$ is a vector representation of $G \ominus v$. As mentioned earlier, this process results in a representation that has rank one less than rank $\overrightarrow{V}$. Unfortunately, $\operatorname{msr}(G) - \operatorname{msr}(G \ominus v)$ may be arbitrarily large as demonstrated by the complete bipartite graph $K_{2,n}$: For $n \geq 3$, by Corollary 2.7, $\operatorname{msr}(K_{2,n}) \geq n$, but the orthogonal removal of a vertex from the smaller independent set yields the complete graph on $n+1$ vertices, K_{n+1}, and $\operatorname{msr}(K_{n+1}) = 1$ by Remark 2.4.

We say that subgraphs $G_1, \ldots, G_m$ [illegible] a graph G with multiple edges if each vertex of G is a vertex of at least one G_i and, for every pair of vertices v and w of G joined by [illegible] one edge, there exists an i such that v and w are joined by [illegible] one edge in G_i. We now restate Lemma 2.8 for graphs with multiple edges.

LEMMA 3.3. [illegible] $X_1, \ldots, X_m$ $X_i \subseteq \mathbb{C}^n$ [illegible] $1 \leq i \leq m$ [illegible] $G_1, \ldots, G_m$ [illegible] G [illegible]

- $G_1, \ldots, G_m$ [illegible] G
- [illegible] v, w [illegible] G, [illegible] $\overrightarrow{x_v}$ [illegible] v [illegible] X_i [illegible] $\overrightarrow{x_w}$ [illegible] w [illegible] X_j $\langle \overrightarrow{x_v}, \overrightarrow{x_w} \rangle = 0$ [illegible] X [illegible] G [illegible]

$$\operatorname{rank} X \leq \operatorname{rank}\left(\bigcup_{1 \leq i \leq m} \operatorname{span} X_i\right) \leq \sum_{1 \leq i \leq m} \operatorname{rank} X_i.$$

Recall that a vertex v such that $n(v)$ induces a complete graph is said to be [illegible]

LEMMA 3.4. [illegible] v [illegible] G [illegible] $\operatorname{msr}(G) = \operatorname{msr}(G \ominus v) + 1$

[illegible]. From Remark 3.2, we have that $\operatorname{msr}(G) \geq \operatorname{msr}(G \ominus v) + 1$. From Remark 2.4, we may find a vector representation of rank one of the subgraph of G induced by v and its neighbors. Choosing this representation to be orthogonal to a representation of $G \ominus v$, we may apply Lemma 3.3 to see that $\operatorname{msr}(G) \leq \operatorname{msr}(G \ominus v) + 1$. ∎

The following corollary simplifies finding the minimum rank of graphs with vertices, which are simply vertices of degree 1. This corollary is also found in [17, Lemma 3.6] with a different proof.

COROLLARY 3.5. $\operatorname{msr}(G) = \operatorname{msr}(G - v) + 1$

A graph is said to be if it has no induced cycles C_n with $n \geq 4$. It is known that every nonempty chordal graph has at least one simplicial vertex [2, p. 175]. A clique cover of a graph G with multiple edges is a collection of cliques of G that cover every edge between the vertices of G. As before, the clique cover number of G $\operatorname{cc}(G)$ is the minimum number of cliques in a clique cover of G. We are now able to show that, for chordal graphs, the msr is the clique cover number.

THEOREM 3.6. $\operatorname{msr}(G) = \operatorname{cc}(G)$

. Induct on the number of vertices of G. We start the induction with an edge. For graphs with three or more vertices, identify a simplical vertex v of G. If, in addition, v is a duplicate vertex, then $\operatorname{cc}(G - v) = \operatorname{cc}(G)$ and $\operatorname{msr}(G - v) = \operatorname{msr}(G)$. If v is not a duplicate vertex and not connected to any other vertex by exactly one edge, then $\operatorname{cc}(G - v) = \operatorname{cc}(G)$ and $\operatorname{msr}(G - v) = \operatorname{msr}(G)$.

Finally, if v is not a duplicate vertex and is connected to at least one other vertex by exactly one edge, we observe that $\operatorname{cc}(G \ominus v) = \operatorname{cc}(G) - 1$. To see this, when v is simplicial, there are multiple edges between each pair of vertices in $n(v)$ in $G \ominus v$. Thus remove exactly one clique from a minimum clique cover of G to obtain a clique cover of $G \ominus v$. Now using Lemma 3.4 we get $\operatorname{msr}(G) = \operatorname{cc}(G)$. □

4. Minimum psd rank for graphs on less than seven vertices. For all the graphs G with $|V(G)| \leq 6$, with a few exceptions listed below, we can determine $\operatorname{msr}(G)$ using results discussed in this paper. A catalog of these graphs can be found in [16]. Table 4.1 lists the minimum psd ranks of 142 graphs on 2 or more vertices but less than seven vertices using the numbering found in [16].

We now detail how to use the results of this paper to find the msr of the graphs listed in Table 4.1. The graphs G174, G175, G198, and G204 are the exceptional cases which cannot be handled by the results presented above. We provide alternate methods for these graphs.

The complete graphs G3, G7, G18, G52, and G208 have msr equal to 1 by Remark 2.4. As mentioned in the introduction, the msr of a tree is one less than the number of vertices. This gives the msr for the trees G3, G6, G13, G14, G29–31, G77–81, and G83.

Among the nontree, noncomplete graphs, the following 64 graphs are chordal: G15, G17, G34–36, G40–42, G45–47, G49, G51, G92–95, G97, G100, G102, G111–

TABLE 4.1

$\operatorname{msr}(G)$	Graph
5	G77–81 and G83.
4	G29–31, G92–100, G102–105, G111-115, G118, G120–125, G127–129, G135–139, G145–149, G152, G161, G162, G164, and G167.
3	G13, G14, G34-38, G40, G41, G43, G44, G46, G47, G117, G119, G126, G130, G133, G134, G140–144, G150, G151, G153, G154, G156–160, G163, G166, G168–175, G177–189, G192, G193, G196–198, G201, and G202.
2	G6, G15–17, G42, G45, G48-51, G165, G190, G191, G194, G195, G199, G200, and G203–207.
1	G3, G7, G18, G52, and G208.

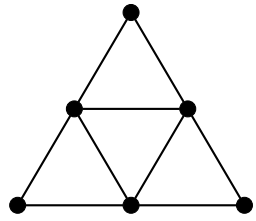


FIG. 4.1. *G*163.

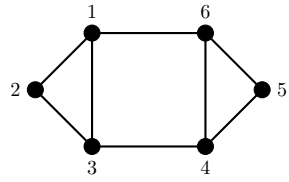


FIG. 4.2. *G*152.

115, G117, G119, G120, G123, G130, G133–139, G142, G144, G150, G156, G157, G160–165, G167, G177–181, G183, G191–193, G195, G200–G202, G205, and G207. Theorem 3.6 gives that the msr of a chordal graph is its clique cover number. For example, we have cc(G163) = msr(G163) = 3 (Figure 4.1).

There are 20 nonchordal graphs whose msr is 4. All but graph G152 (Figure 4.2) satisfy $\mathrm{ts}(G) = 5$. The discussion following Lemma 2.5 shows that, for these graphs, $\mathrm{msr}(G) = 4$. For G152, if we orthogonally remove simplicial vertices 2 and 5 and apply Lemma 3.4, we observe that msr(G152) = 4. In addition, G152 is not chordal, but $\mathrm{msr}(G) = \mathrm{cc}(G) = 4$, indicating that the converse to Theorem 3.6 is false.

Among the 32 nonchordal graphs whose msr is 3, G37, G38, G43, and G44 have $\mathrm{ts}(G) = 4$, hence they have $\mathrm{msr}(G) = 3$. The msr of G140, G141, G143, G158, and G159 is 3 by Corollary 3.5. A duplicate vertex is removed in G126, G153, G168, G169, G170, G172, G185, and G189, and the resulting graph on 5 vertices has msr equal to 3. The graphs G151, G154, G166, G171, G173, G182, G184, G186–G188, G196, and G197 satisfy the sufficient condition of Theorem 2.9. The exceptional cases are G174, G175, and G198. These three graphs could be handled using a construction as shown below or by applying Theorem 3.1 and Proposition 3.2 of [17] along with Lemma 2.5.

A maximum induced tree of G198 (Figure 4.3) is induced by $\{v_1, v_2, v_3, v_4\}$. Using the Laplacian matrix of this tree in the top left 4-by-4 block, we construct rows 5 and 6 to represent the graph G198,

$$\begin{bmatrix} 1 & -1 & 0 & 0 & -1 & 1 \\ -1 & 2 & -1 & 0 & 1 & -1 \\ 0 & -1 & 2 & -1 & 1 & 1 \\ 0 & 0 & -1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 & 2 & 0 \\ 1 & -1 & 1 & -1 & 0 & 2 \end{bmatrix}.$$

The graph G198, with this rank 3 psd matrix, is an example which shows that the $\circledast$ condition of Theorem 2.9 is not necessary.

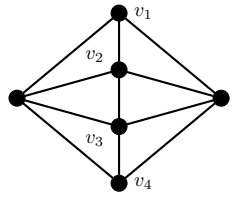


FIG. 4.3. $G198$.

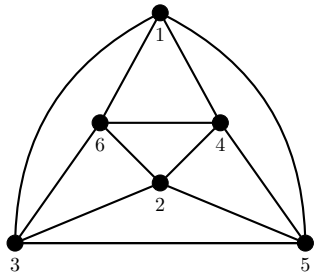


FIG. 4.4. $G204$.

Among the 9 nonchordal graphs whose msr is 2, G16 is a cycle on 4 vertices, while G50 and G203 satisfy the sufficient condition of Theorem 2.9. Removing one duplicate vertex from G48 and G206, and removing two duplicate vertices from G190, G194 and G199 reduce the graph to a known case. The one exceptional case is G204 (see Figure 4.4).

Suppose $\overrightarrow{e_1} = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ and $\overrightarrow{e_2} = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$. Then we can write a vector representation for G204 as follows: $\overrightarrow{v_1} = \overrightarrow{e_1}$, $\overrightarrow{v_2} = \overrightarrow{e_2}$, $\overrightarrow{v_3} = 2\overrightarrow{e_1} + \overrightarrow{e_2}$, $\overrightarrow{v_4} = \overrightarrow{e_1} - 2\overrightarrow{e_2}$, $\overrightarrow{v_5} = \overrightarrow{e_1} + \overrightarrow{e_2}$, and $\overrightarrow{v_6} = \overrightarrow{e_1} - \overrightarrow{e_2}$. Thus msr(G204) = 2. Alternatively, we may use [1, Theorem 15].

Acknowledgments. The authors thank one of the referees for suggesting a shorter approach to the proof of Theorem 3.6. B. Harris, M. Lay, S. Narayan, A. Pascoe, K. Steinmetz, and W. Wang participated in an NSF–REU program at Central Michigan University during the summers of 2003 and 2004. M. Booth, C. Johnson, and B. Sutton participated in an NSF–REU program at the College of William and Mary in 2003.

REFERENCES

[1] W. BARRETT, H. VAN DER HOLST, AND R. LOEWY, *Graphs whose minimal rank is two*, Electron. J. Linear Algebra, 11 (2004), pp. 258–280.
[2] B. BOLLOBÁS, *Modern Graph Theory*, Grad. Texts in Math. 184, Springer, New York, 1984.
[3] Y. COLIN DE VERDIÈRE, *Multiplicities of eigenvalues and tree-width of graphs*, J. Combin. Theory Ser. B, 74 (1998), pp. 121–146.
[4] P. ERDÖS, M. SAKS, AND V. SÓS, *Maximum induced trees in graphs*, J. Combin. Theory Ser. B, 41 (1986), pp. 61–79.
[5] R. HORN AND C.R. JOHNSON, *Matrix Analysis*, Cambridge University Press, New York, 1985.
[6] C. R. JOHNSON AND A. LEAL-DUARTE, *The maximum multiplicity of an eigenvalue in a matrix whose graph is a tree*, Linear Multilinear Algebra, 46 (1999), pp. 139–144.
[7] C. R. JOHNSON AND A. LEAL-DUARTE, *On the possible multiplicities of the eigenvalues of a Hermitian matrix whose graph is a tree*, Linear Algebra Appl., 348 (2002), pp. 7–21.
[8] C. R. JOHNSON AND A. LEAL-DUARTE, *Converse to the Parter-Wiener theorem: The case of non-trees*, Discrete Math., 306 (2006), pp. 3125–3129.

[9] C. R. Johnson and C. M. Saiago, *Estimation of the maximum multiplicity of an eigenvalue in terms of the vertex degrees of the graph of a matrix*, Electron. J. Linear Algebra, 9 (2002), pp. 27–31.
[10] C. R. Johnson, A. Leal-Duarte, C. M. Saiago, B. D. Sutton, and A. J. Witt, *On the relative position of multiple eigenvalues in the spectrum of an Hermitian matrix with a given graph*, Linear Algebra Appl., 363 (2003), pp. 147–159.
[11] C. R. Johnson, A. Leal-Duarte, and C. M. Saiago, *Inverse eigenvalue problems and lists of multiplicities of eigenvalues for matrices whose graph is a tree: The case of generalized stars and double generalized stars*, Linear Algebra Appl., 373 (2003), pp. 311–330.
[12] C. R. Johnson, B. Kroschel, and M. Omladič, *Eigenvalue multiplicities in principal submatrices*, Linear Algebra Appl., 390 (2004), pp. 111–120.
[13] A. Leal-Duarte and C. R. Johnson, *On the minimum number of distinct eigenvalues for a symmetric matrix whose graph is a given tree*, Math. Inequal. Appl., 5 (2002), pp. 175–180.
[14] R. Merris, *A survey of graph Laplacians*, Linear Multilinear Algebra, 39 (1995), pp. 19–31.
[15] N. J. Pullman, *Clique Coverings of Graphs—A survey*, in Combinatorial Mathematics X, L. R. A. Casse, ed., Springer, Berlin, 1983, pp. 72–85.
[16] R. C. Read and R. J. Wilson, *An Atlas of Graphs*, Oxford University Press, New York, 1998.
[17] H. van der Holst, *Graphs whose positive semi-definite matrices have nullity at most two*, Linear Algebra Appl., 375 (2003), pp. 1–11.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 741–761

MATRIX VALUED ORTHOGONAL POLYNOMIALS ARISING FROM GROUP REPRESENTATION THEORY AND A FAMILY OF QUASI-BIRTH-AND-DEATH PROCESSES*

F. ALBERTO GRÜNBAUM† AND MANUEL D. DE LA IGLESIA‡

Abstract. We consider a family of matrix valued orthogonal polynomials obtained by Pacharoni and Tirao in connection with spherical functions for the pair $(\mathrm{SU}(N+1), \mathrm{U}(N))$; see [I. Pacharoni and J. A. Tirao, *Constr. Approx.*, 25 (2007), pp. 177–192]. After an appropriate conjugation, we obtain a new family of matrix valued orthogonal polynomials where the corresponding block Jacobi matrix is stochastic and has special probabilistic properties. This gives a highly nontrivial example of a nonhomogeneous quasi-birth-and-death process for which we can explicitly compute its "n-step transition probability matrix" and its invariant distribution. The richness of the mathematical structures involved here allows us to give these explicit results for a several parameter family of quasi-birth-and-death processes with an arbitrary (finite) number of phases. Some of these results are plotted to show the effect that choices of the parameter values have on the invariant distribution.

Key words. matrix valued orthogonal polynomials, Markov chains, block tridiagonal transition matrix, quasi-birth-and-death processes

AMS subject classifications. 60J10, 42C05

DOI. 10.1137/070697604

1. Purpose and contents of the paper. The aim of this paper is to tie together two subjects that have received quite a bit of attention recently. We will not give a detailed explanation of either one of them, since this would require too much space and it has been done properly in the literature already. Besides, since these two subjects require rather different backgrounds, an ab-initio exposition would be a formidable task. To compensate for this we give a brief historical view of how these topics developed and then combine them at the appropriate point. The contents of this paper can be divided into three parts.

A first part gives a brief account of the subjects that are going to play a role in this paper. The introduction contains some historical developments tying the moment problem with spectral theory and a quick look at birth-and-death processes, including the appearance of the appropriate orthogonal polynomials. Section 3 reviews very briefly Kreĭn's theory of matrix valued orthogonal polynomials and discusses the first example relevant to our considerations. Section 4 gives a minimal description of the class of Markov chains known as quasi-birth-and-death processes and talks about the very natural connection between this and the previous section.

The second part introduces the family of examples arising from group representation theory that we are going to use in this paper. Section 5 gives a guide to the literature on matrix valued spherical functions aimed at showing how the examples discussed in section 6 arose. Section 6 gives the bare-bones details of the extensive

*Received by the editors July 18, 2007; accepted for publication (in revised form) by H. J. Woerdeman March 14, 2008; published electronically July 2, 2008.
http://www.siam.org/journals/simax/30-2/69760.html

†Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720 (grunbaum@math.berkeley.edu). This author's work was partially supported by NSF grant DMS-0603901.

‡Departamento de Análisis Matemático, Universidad de Sevilla, Apdo (P.O. BOX) 1160, 41080 Sevilla, Spain (mdi29@us.es). This author's work was partially supported by D.G.E.S., ref. BFM2003-06335-C03-01, FQM-262 P06-FQM-01735, FQM-481 (Junta de Andalucía).

work carried out in [30] in the case of the complex projective space. Section 7 shows how to conjugate the weight matrix from [30] to obtain a family of matrix valued orthogonal polynomials with extra probabilistic properties.

The third part concentrates on the probabilistic aspects of our family of examples. Section 8 deals with a number of issues of probabilistic nature and displays the network associated with our examples. In particular we find explicit expressions for the invariant distribution. Section 9 gives graphical displays of some results obtained from the exact formulas in the previous sections. The goal here is to show that by varying the parameters afforded by the group theoretical situation one can obtain quite a range of different probabilistic behaviors. Finally, section 10 gives a summary of the results of the paper and the challenges that lie ahead.

2. Introduction. The classical Hausdorff moment problem, that of determining a measure $d\psi(x)$ in the interval $[-1, 1]$ from its moments

$$\sigma_n = \int_{-1}^{1} x^n d\psi(x),$$

originated in very concrete problems at the end of the 19th century and was discussed by people such as Chebyshev, Markov, and Stieltjes. In the hands of Weyl and a few others, this showed the far reaching power of the modern theory of functional analysis in the early part of the 20th century. The main ingredient here is to connect this problem with the spectral theory of a second order difference operator (built from the moments σ_n) acting on functions defined on the nonnegative integers. The moments in question determine (up to scalars) a family of polynomials $\{Q_n(x)\}_{n\geq 0}$, and these polynomials are the eigenfunctions of the second order difference operator alluded to above. In the appropriate Hilbert space this operator is symmetric, and the problem of finding $d\psi(x)$ is the problem of finding self-adjoint extensions of this symmetric operator. Under certain conditions there is a unique such extension and thus a unique solution to the moment problem we started from, but at any rate any extension gives a measure $d\psi(x)$ that makes the polynomials orthogonal with respect to each other.

To get closer to our subject we need a few more ingredients. One of them is given in the rest of this section, and the other two in sections 3 and 4.

The presence of a second order difference operator acting on the space of functions defined on the nonnegative integers, i.e., a semi-infinite tridiagonal matrix, makes it natural to think of a very special kind of Markov chain on the space of nonnegative integers. These are the so-called birth-and-death processes where at each discrete unit of time a transition is allowed from state i to state j with probability P_{ij} and we put $P_{ij} = 0$ if $|i - j| > 1$. The one-step transition probability matrix is given by

$$P = \begin{pmatrix} r_0 & p_0 & & & \\ q_1 & r_1 & p_1 & & \\ & q_2 & r_2 & p_2 & \\ & & \ddots & \ddots & \ddots \end{pmatrix}. \tag{2.1}$$

We will assume that $p_j > 0$, $q_{j+1} > 0$, and $r_j \geq 0$ for $j \geq 0$. We also assume $p_j + r_j + q_j = 1$ for $j \geq 1$ and by putting $p_0 + r_0 \leq 1$ we allow for the state $j = 0$ to be an absorbing state (with probability $1 - p_0 - r_0$). Some of these conditions can be relaxed.

The problem here is to obtain an expression for the so-called "n-step transition probability matrix," giving the probability of going between any two states in n steps. By making use of the ideas mentioned above, i.e., by bringing in an appropriate Hilbert space and applying then the spectral theorem, Karlin and McGregor [20] obtained a neat representation formula for the quantity of interest, as recalled below.

If one introduces the polynomials $\{Q_n(x)\}_{n\geq 0}$ by the conditions $Q_{-1}(x) = 0$, $Q_0(x) = 1$ and by using the notation

$$\phi = \begin{pmatrix} Q_0(x) \\ Q_1(x) \\ \vdots \end{pmatrix},$$

one insists on the recursion relation

$$P\phi = x\phi,$$

it is possible to prove the existence of a unique measure $d\psi(x)$ supported in $[-1, 1]$ such that

$$\int_{-1}^{1} Q_i(x)Q_j(x)d\psi(x) \Big/ \int_{-1}^{1} Q_j(x)^2 d\psi(x) = \delta_{ij}$$

and one gets the Karlin–McGregor representation formula

$$P_{ij}^n = \int_{-1}^{1} x^n Q_i(x)Q_j(x)d\psi(x) \Big/ \int_{-1}^{1} Q_j(x)^2 d\psi(x). \tag{2.2}$$

If time is taken to be continuous, as it is done in other papers by Karlin and McGregor, then this formula and the matrix P suffer only cosmetic changes.

It is interesting to notice that this seminal paper of Karlin and McGregor refers both to the standard text on the moment problem at the time [33], as well as to the fact that Feller and McKean had already recognized the relevance of the Hilbert space setup in the study of diffusion processes; see [7, 26]. One can mention other papers, such as [8, 17, 19, 25], where similar ideas were at play.

The last section of [20] deals with the case of a finite state space and the case when the nonnegative integers are replaced by the set of all integers. Since one is using a very powerful tool such as the spectral theorem it is clear that an adaptation of the ideas from birth-and-death processes will work here too. In the case of the integers, one is dealing with a state space with two singular points (one at each end of the line), and in this case Weyl and others had already found the correct tool: one replaces the spectral measure $d\psi(x)$ by a 2×2 nonnegative matrix. The paper of Karlin and McGregor concludes with the explicit computation of this matrix in the case of the doubly infinite random walk. The general formula is given in expression (12) of [20] for the case of discrete time and also in (6.8) of [18] for continuous time.

The representation formula given above is of intrinsic interest: the computation of the left-hand side of (2.2) for fixed i, j and arbitrary values of n involves all of the entries of (2.1). However, if $d\psi(x)$ is known, then the right-hand side of (2.2) gives a way of computing this quantity using only a fixed number of entries of (2.1).

The applicability of (2.2) depends to a large extent on our ability to obtain useful expressions for the orthogonal polynomials and the orthogonality measure associated

with P. If one looks around in the literature one discovers that the number of cases where this is possible is rather small.

We close this section by showing how one can compute in the case of a stochastic matrix P its invariant (stationary) distribution, i.e., the (unique up to scalars) row vector

$$\boldsymbol{\pi} = (\pi_0, \pi_1, \pi_2, \dots)$$

such that

$$\boldsymbol{\pi} P = \boldsymbol{\pi}.$$

Recall that a matrix P with nonnegative entries is called stochastic if the sum of the elements in any row equals unity.

We first obtain, using $r_0 + p_0 = 1$, that $\pi_1 = \pi_0 p_0/q_1$. Then one proves by induction that for $i \geq 1$ we have

$$\pi_i = \pi_0(p_0 p_1 \dots p_{i-1})/(q_1 q_2 \dots q_i).$$

This has the consequence that

$$\pi_{i+1}/\pi_i = p_i/q_{i+1}.$$

Now for $i \geq 0$ we have

$$xQ_i(x) = p_i Q_{i+1}(x) + r_i Q_i(x) + q_i Q_{i-1}(x)$$

with $q_0 = 0$. Integrating this after multiplication by Q_{i+1} or Q_{i-1} gives

$$\int_{-1}^{1} xQ_i(x)Q_{i+1}(x)d\psi(x) = p_i \int_{-1}^{1} Q_{i+1}^2(x)d\psi(x) = q_{i+1} \int_{-1}^{1} Q_i^2(x)d\psi(x).$$

Combining these two results we get that the ratio of the two integrals above is given by the common value

$$q_{i+1}/p_i = \pi_i/\pi_{i+1}.$$

The moral of this is that the solution to $\boldsymbol{\pi} P = \boldsymbol{\pi}$ can be computed (up to a common multiplicative scalar) either from the matrix P itself or from the knowledge of the integrals

$$\int_{-1}^{1} Q_i^2(x)d\psi(x).$$

In particular if we have an homogeneous birth-and-death process where $p_i = p$ and $q_i = q$ independently of the value of i, then we have that the components of $\boldsymbol{\pi}$ are given by $\pi_i = \pi_0(p/q)^i$, $i \geq 0$.

3. Matrix valued orthogonal polynomials. We need two more characters to be able to start our tale. The first one is the theory of matrix valued orthogonal polynomials, whose bare-bones development is given in two papers by Kreĭn [21, 22]. There is no written account of the motivation that led Kreĭn to this theory, but one can easily see the connection with the spectral theory of difference operators on the integers. This is very nicely discussed in the book by Berezans'kiĭ [2]. In fact the study

of the classical second order difference operator on the integers is done in detail in [2] and may constitute the first example of the theory of Kreĭn, where the polynomials and their orthogonality weight matrix $W(x)$ are both explicitly given. This is, of course, a special case of the matrix that appears in the last section of [20] for general values of p and q $(p+q=1)$.

We give now a brief account of Kreĭn's theory.

Given a positive definite matrix valued measurable weight function $W = W(x)$ with finite moments we can consider the skew symmetric bilinear form defined for any pair of matrix valued polynomial functions $P(x)$ and $Q(x)$ by the numerical matrix

$$(P,Q) = (P,Q)_W = \int_{\mathbb{R}} P(x)W(x)Q^*(x)dx,$$

where $Q^*(x)$ denotes the conjugate transpose of $Q(x)$. We define the matrix valued norm of P by

$$\|P\|^2 = (P,P)_W. \tag{3.1}$$

One can also deal with a more general weight matrix $W(x)$; see [4].

This leads, using the Gram–Schmidt process, to the existence of a sequence of matrix valued orthogonal polynomials with nonsingular leading coefficients. Given an orthogonal sequence $\{Q_n(x)\}_{n\geq 0}$ one gets a three term recursion relation

$$xQ_n(x) = A_nQ_{n+1}(x) + B_nQ_n(x) + C_nQ_{n-1}(x), \tag{3.2}$$

where A_n is nonsingular. We will denote by $\mathcal{L}$ the corresponding [illegible], defined by the following block tridiagonal semi-infinite matrix:

$$\mathcal{L} = \begin{pmatrix} B_0 & A_0 & & & \\ C_1 & B_1 & A_1 & & \\ & C_2 & B_2 & A_2 & \\ & & \ddots & \ddots & \ddots \end{pmatrix}.$$

Using the notation

$$\Phi = \begin{pmatrix} Q_0(x) \\ Q_1(x) \\ \vdots \end{pmatrix}$$

the relation (3.2) becomes

$$\mathcal{L}\Phi = x\Phi. \tag{3.3}$$

We will reserve the symbol P for the case where $\mathcal{L}$ becomes a one-step transition probability matrix, thought of as a scalar matrix. The corresponding Markov chain (to appear in section 4) will have a state space that is more complicated than the set $\{0,1,2,\dots\}$ corresponding to a birth-and-death process featured in section 2.

In the scalar case, concrete examples of orthogonal polynomials, including explicit formulas for them as well as their orthogonality measure preceded the development of any general theory. Prominent examples are the Hermite, Laguerre, and Jacobi

polynomials. These examples arose from concrete problems in the eighteenth and nineteenth centuries and played a fundamental role, in the hands of Schrödinger, in the development of quantum mechanics around 1925.

The situation in the matrix valued case is entirely different: the general theory just described above came first. Until a few years ago, it may be that the only nontrivial example was the one included in Berezans'kiĭ's book [2], alluded to above and recalled below for the benefit of the reader.

Consider the block tridiagonal matrix

$$\mathcal{L} = \begin{pmatrix} B_0 & I & & & \\ C_1 & B_1 & I & & \\ & C_2 & B_2 & I & \\ & & \ddots & \ddots & \ddots \end{pmatrix}$$

with 2×2 blocks given as follows:

$$B_0 = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad B_n = 0 \quad \text{if } n \geq 1,$$

$$C_n = \frac{1}{4} I \quad \text{if } n \geq 1,$$

where I stands for the identity matrix. In this case one can compute explicitly the matrix valued polynomials $\{Q_n(x)\}_{n\geq 0}$ given by

$$xQ_n(x) = Q_{n+1}(x) + B_n Q_n(x) + C_n Q_{n-1}(x), \quad Q_{-1}(x) = 0, \quad Q_0(x) = I.$$

One gets

$$Q_n(x) = \frac{1}{2^n}\begin{pmatrix} U_n(x) & -U_{n-1}(x) \\ -U_{n-1}(x) & U_n(x) \end{pmatrix},$$

where $U_n(x)$ are the Chebyshev polynomials of the second kind.

The orthogonality measure is read off from the identity

$$\frac{4^i}{\pi}\int_{-1}^{1} Q_i(x) \frac{1}{\sqrt{1-x^2}} \begin{pmatrix} 1 & x \\ x & 1 \end{pmatrix} Q_j^*(x)dx = \delta_{ij} I.$$

Proceeding as in [3, 10, 20] one obtains a Karlin–McGregor representation. We get, for $n = 0, 1, 2, \ldots,$

$$\mathcal{L}_{ij}^n = \frac{4^i}{\pi}\int_{-1}^{1} x^n Q_i(x) \frac{1}{\sqrt{1-x^2}} \begin{pmatrix} 1 & x \\ x & 1 \end{pmatrix} Q_j^*(x)dx,$$

where $\mathcal{L}_{ij}^n$ stands for the (i, j) block of the matrix $\mathcal{L}^n$. As is usual for birth-and-death processes, the indices i, j run from 0 on.

In this way, as noticed in [10], one can compute the entries of the powers $\mathcal{L}^n$ with

$\mathcal{L}$ thought of as a pentadiagonal scalar matrix, namely

$$\mathcal{L} = \begin{pmatrix} 0 & \frac{1}{2} & 1 & 0 & & \\ \frac{1}{2} & 0 & 0 & 1 & 0 & \\ \frac{1}{4} & 0 & 0 & 0 & 1 & \ddots \\ & \frac{1}{4} & 0 & 0 & 0 & \ddots \\ & & \ddots & \ddots & \ddots & \ddots \end{pmatrix}.$$

$\mathcal{L}$ is not a stochastic matrix since its rows do not add up to unity. Nevertheless, defining Δ to be the 2×2 block diagonal matrix with $\Delta_{ii} = 2^i I$ for every block, we get from (3.3) that $\Delta\mathcal{L}\Delta^{-1}\Delta\Phi = x\Delta\Phi$ and thus if $P = \Delta\mathcal{L}\Delta^{-1}$ and $\widetilde{\Phi} = \Delta\Phi$, we have $P\widetilde{\Phi} = x\widetilde{\Phi}$. The scalar version of P is now the stochastic matrix

$$P = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & & \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & \ddots \\ & \frac{1}{2} & 0 & 0 & 0 & \ddots \\ & & \ddots & \ddots & \ddots & \ddots \end{pmatrix}.$$

Observe that the norm of $\widetilde{Q}_n$, defined in (3.1), satisfies $\|\widetilde{Q}_n\|^2 = \pi$. This is nothing but the example considered at the end of [20] in the special case of $p = q = 1/2$.

In the last few years a number of new families of matrix valued orthogonal polynomials have been computed explicitly along with their orthogonality measure. Typically they are joint eigenfunctions of some fixed differential operator with matrix coefficients. This search was initiated in [5], but nontrivial examples were not discovered until [6] and [13, 16]. The family of examples we will consider later is related to one of these examples and is obtained by modifying the situation discussed in [30].

4. Quasi-birth-and-death processes. Now we come to the last character of our story. For our purposes, we consider a two dimensional Markov chain with discrete time. The state space consists of the pair of integers (i, j), $i \in \{0, 1, 2, \dots\}$, $j \in \{1, \dots, d\}$. The first component is usually called the level and the second one the phase. The one-step transition probability matrix, which we will denote (as before) by P has a block tridiagonal structure (see (4.2)). This indicates that in one unit of time a transition can change the phase without changing the level, or can change the level (and possibly the phase) to either of the adjacent levels. The probability of going in one step from state (i, j) to state (i', j') is given by the (j, j') element of the block $P_{i,i'}$. Clearly in the case when the number of phases d is one we are back to the case of an ordinary birth-and-death process. In general, these processes are known as (discrete time) quasi-birth-and-death processes.

For a much more detailed presentation of this field, as well as its connections with queueing problems in network theory as well as the general field of communication systems, the reader should consult [24, 27] and some of the references in [3].

Once one has the notions introduced in the previous sections it is very natural to connect them and to analyze these interesting Markov chains in terms of the

corresponding spectral properties of the resulting family of matrix valued orthogonal polynomials. As pointed out before one can see the seed for this in the work of Kreĭn, as well as in the original paper [20].

We have identified two references where the corresponding Karlin–McGregor representation formula has been explicitly given, but there may be others since this is such a natural extension of the scalar theory; see [3] and [10]. In the case of quasi-birth-and-death processes one replaces (2.2) by

$$P_{ij}^n = \left(\int x^n Q_i(x) W(x) Q_j^*(x) dx \right) \left(\int Q_j(x) W(x) Q_j^*(x) dx \right)^{-1}. \tag{4.1}$$

A different but related path to this circle of ideas in connection with network models can be seen in [1].

In [3] one finds some interesting examples where this representation formula is computed explicitly, including a new derivation of the result dealing with the case of random walk on the integers. In [10] one finds an example, taken from [9], where the observation is made that one has a stochastic matrix. The family of examples to be considered in the following sections is an extension of this example.

Given the block transition probability matrix P, the problem of computing an invariant distribution row vector, i.e., a vector with nonnegative entries π_j^i,

$$\boldsymbol{\pi} = (\boldsymbol{\pi}^0; \boldsymbol{\pi}^1; \dots) \equiv (\pi_1^0, \pi_2^0, \dots, \pi_d^0; \pi_1^1, \pi_2^1, \dots, \pi_d^1; \dots)$$

such that

$$\boldsymbol{\pi} P = \boldsymbol{\pi}$$

leads to a complicated system of equations.

If

$$P = \begin{pmatrix} B_0 & A_0 & & & \\ C_1 & B_1 & A_1 & & \\ & C_2 & B_2 & A_2 & \\ & & \ddots & \ddots & \ddots \end{pmatrix}, \tag{4.2}$$

we have

$$\boldsymbol{\pi}^0 B_0 + \boldsymbol{\pi}^1 C_1 = \boldsymbol{\pi}^0$$

and then, for $n \geq 1$,

$$\boldsymbol{\pi}^{n-1} A_{n-1} + \boldsymbol{\pi}^n B_n + \boldsymbol{\pi}^{n+1} C_{n+1} = \boldsymbol{\pi}^n.$$

This gives, as is easy to check, the rather unpleasant expressions

$$\boldsymbol{\pi}^1 = \boldsymbol{\pi}^0 (I - B_0) C_1^{-1},$$

$$\boldsymbol{\pi}^2 = \boldsymbol{\pi}^0 [(I - B_0) C_1^{-1} (I - B_1) - A_0] C_2^{-1},$$

$$\boldsymbol{\pi}^3 = \boldsymbol{\pi}^0 [(I - B_0) C_1^{-1} (I - B_1) C_2^{-1} (I - B_2) - A_0 C_2^{-1} (I - B_2) - (I - B_0) C_1^{-1} A_1] C_3^{-1}.$$

These formulas require that the matrices C_n be invertible. Under these conditions, one can derive nicer looking expressions for the invariant distribution (see [23]). There are

many issues here that we leave untouched in this analysis. For instance, the possibility of choosing $\boldsymbol{\pi}^0$ with nonnegative entries so that all the subsequent $\boldsymbol{\pi}^n$ will have this property requires extra conditions.

The general theory of quasi-birth-and-death processes is not restricted to the case when the matrices A_n, B_n, and C_n are all square matrices of the same size and A_n and C_n are nonsingular. It remains an interesting challenge to find a mathematical setup that can accommodate such a situation.

Now that we have seen how to extend the Karlin–McGregor representation formula to the block tridiagonal case it is important to get examples where the polynomials and the orthogonality matrix can be explicitly written down. This is the purpose of the next three sections.

5. Matrix valued spherical functions and matrix valued orthogonal polynomials. The theory of matrix valued spherical functions, initially discussed in [34], is one of the routes leading to explicit families of matrix valued orthogonal polynomials and their orthogonality measure. The first family of examples appeared in [13, 16] in connection with $G = \mathrm{SU}(3)$. The size of the matrices here is already arbitrary, and the orthogonality matrix has a scalar factor of the form $x^\alpha(1-x)$. The extension to the case where this scalar factor can be taken to be $x^\alpha(1-x)^\beta$ for arbitrary $\alpha, \beta > -1$ was undertaken in [9] in the 2×2 case without any reference to group representation theory. Further examples of this kind are given in [14]. The role of group representation theory in getting away from the 2×2 case can be seen in [15]. Finally, [30] displays for the case of $G = \mathrm{SU}(N+1)$ families of orthogonal polynomials depending on three parameters, α, β, and k. In the special case of $k = \frac{\beta+1}{2}$, one recovers the results of [9].

These orthogonal polynomials are given by properly "packaged and conjugated" sets of matrix valued spherical functions. These spherical functions correspond to irreducible representations of $\mathrm{U}(N)$ and therefore are parameterized by partitions

$$\mu = (m_1, m_2, \ldots, m_N) \in \mathbb{Z}^N \quad \text{such that} \quad m_1 \geq m_2 \geq \cdots \geq m_N.$$

In this paper, following [30], we use only "one step" representations given by a partition

$$\mu = (\underbrace{m+\ell, \ldots, m+\ell}_{k}, \underbrace{m, \ldots, m}_{N-k}), \quad 1 \leq k \leq N-1.$$

In terms of the parameters α and β, one has $\alpha = m$ and $\beta = N-1$. The remaining free parameter ℓ will determine the size of the corresponding matrix valued orthogonal polynomials and is independent of N. In the next section it will be related to the parameter d appearing in [3].

The examples that have been worked out so far indicate that the matrix valued orthogonal polynomials that result from matrix valued spherical functions lead to a block tridiagonal matrix that can be made into a stochastic one. This will be seen, for our family of examples, in section 8.

Although it is possible to obtain examples of stochastic matrices arising in a different fashion, see, for instance [2], we are not aware of any other general scheme that would produce these desirable kinds of matrices in a systematic fashion.

6. A family of examples arising from the complex projective space. In what follows we shall use E_{ij} to denote the matrix with entry (i,j), which is equal to 1 and 0 elsewhere, where the indices i, j run from 0 on.

Let $d \in \{1, 2, 3, \dots\}$, $\alpha, \beta > -1$ and $0 < k < \beta + 1$. From [30] we have that the differential operator

$$D = x(1-x)\frac{d^2}{dx^2} + (C - xU)\frac{d}{dx} + V,$$

with matrix coefficients given by

$$C = \sum_{i=0}^{d-1}(\alpha + d - i)E_{ii} - \sum_{i=0}^{d-2}(i+1)E_{i+1,i}, \quad U = \sum_{i=0}^{d-1}(\alpha + \beta + d + i + 1)E_{ii},$$

$$V = -\sum_{i=0}^{d-1} i(\alpha + \beta - k + i + 1)E_{ii} + \sum_{i=0}^{d-2}(d - i - 1)(\beta - k + i + 1)E_{i,i+1},$$

admits as eigenfunctions (the conjugates of) a family of orthogonal polynomials with respect to the $d \times d$ weight function

$$W(x) = x^{\alpha}(1-x)^{\beta}Z(x), \quad x \in [0, 1],$$

where

$$Z(x) = \sum_{i,j=0}^{d-1}\left(\sum_{r=0}^{d-1}\binom{r}{i}\binom{r}{j}\binom{d+k-r-2}{d-r-1}\binom{\beta-k+r}{r}(1-x)^{i+j}x^{d-r-1}\right)E_{ij}.$$

Initially the value of k is an integer, but (with the appropriate notation) this can be taken to be any value in the range indicated above. Likewise, in the group theoretical setup of [30] one first takes α and β to be integers but then observes that this holds for α and β as above.

In the language of [14], $\{W, D\}$ is a [illegible].

In [28] one finds an explicit expression for a family of eigenfunctions of D in terms of the matrix valued hypergeometric function ${}_2F_1$ introduced in [35].

In principle it is possible to obtain the coefficients for the three term recursion relation satisfied by any family of orthogonal polynomials whose existence is proved in [30]. We have not done this, but instead, since our goal is to obtain a family of matrix valued orthogonal polynomials $\{Q_n(x)\}_{n\geq 0}$ with specific probabilistic properties, we modify appropriately the classical pair in [30]. This is the goal of the next section.

[illegible] Notice that the notation in [13, 14, 15, 16, 30] and the one in [3] (which we follow here) are related by $d = \ell + 1$.

7. The new equivalent classical pair. Let us consider the following nonsingular upper triangular matrix:

$$T = \sum_{i\leq j}(-1)^i \frac{(-j)_i}{(1-d)_i}\frac{(\alpha + \beta - k + j + 1)_i}{(\beta - k + 1)_i}E_{ij},$$

where $(a)_n$ will denote the Pochhammer symbol defined by $(a)_n = a(a+1)\cdots(a+n-1)$ for $n > 0$, $(a)_0 = 1$. The purpose of choosing T as above will be made clear below.

Let us consider the new classical pair $\{\widetilde{W}, \widetilde{D}\}$, where

$$\widetilde{W} = T^* W T$$

and

$$\widetilde{D} = T^{-1}DT = x(1-x)\frac{d^2}{dx^2} + (\widetilde{C} - x\widetilde{U})\frac{d}{dx} + \widetilde{V},$$

where

$$\begin{aligned}
\widetilde{C} = &\sum_{i=0}^{d-1}\left(\alpha + d - i + \frac{i(d-i)(\beta-k+i)}{\alpha+\beta-k+2i} - \frac{(i+1)(d-i-1)(\beta-k+i+1)}{\alpha+\beta-k+2i+2}\right)E_{ii} \\
&+ \sum_{i=0}^{d-2}\left(1 + i + \frac{(i+1)(d-i-2)(\beta-k+i+2)}{\alpha+\beta-k+2i+3}\right. \\
&\qquad\left. - \frac{(i+1)(d-i-1)(\beta-k+i+1)}{\alpha+\beta-k+2i+2}\right)E_{i,i+1} \\
&+ \sum_{i=0}^{d-2}\left(\frac{(i+1)(d-i-1)(\beta-k+i+1)}{\alpha+\beta-k+2i+2} - \frac{i(d-i-1)(\beta-k+i+1)}{\alpha+\beta-k+2i+1}\right)E_{i+1,i},
\end{aligned}$$

$$\widetilde{U} = \sum_{i=0}^{d-1}(\alpha+\beta+d+i+1)E_{ii} + \sum_{i<j}\left((-1)^{j-i}(i+1)_{j-i}\frac{\alpha+\beta-k+2i+1}{(\alpha+\beta-k+i+1)_{j-i}}\right)E_{ij},$$

$$\widetilde{V} = -\sum_{i=0}^{d-1} i(\alpha+\beta-k+i+1)E_{ii}.$$

The pair $\{\widetilde{W}, \widetilde{D}\}$ is equivalent to $\{W, D\}$ according to the definitions in [14]. Note that T is chosen so that $\widetilde{V}$ turns out to be diagonal.

We now produce a particular family of polynomials $\{Q_n(x)\}_{n\geq 0}$ satisfying $Q_0(x) = I$ and $\widetilde{D}Q_n^*(x) = Q_n^*(x)\Lambda_n$ with

$$\Lambda_n = -\sum_{i=0}^{d-1}(n^2 + (\alpha+\beta+d+i)n + i(\alpha+\beta-k+i+1))E_{ii}.$$

If we put $Q_n(x) = \sum_{j=0}^{n} A_j^n x^j$, we see that the leading coefficient A_n^n can be conveniently chosen to be the lower triangular matrix

$$\begin{aligned}
&\sum_{i\leq j}(-1)^{i+j}\binom{j}{i}(\alpha+\beta-k+2i+1) \\
&\times \frac{(n)_{j-i}(k+n)_{d-j-1}(\alpha+\beta-k+n+j+1)_i(\alpha+\beta+n+d+j)_n}{(k)_{d-j-1}(\beta+d)_n(\alpha+\beta-k+i+1)_{j+1}}E_{ji},
\end{aligned}$$

and that this determines A_j^n for $j = n-1, n-2, \ldots, 0$. This is very similar to the analysis in section 3.1 of [11], where the final result is equations (3)–(5), expressing these polynomials in terms of the matrix valued hypergeometric function ${}_2F_1$ of Tirao; see [35]. As mentioned in the last section, such an expression in terms of ${}_2F_1$ is obtained in [28] for a family of polynomials that are related to $\{Q_n(x)\}_{n\geq 0}$ in the form $Q_n(x) = A_n^n T^*(\widetilde{A}_n^n)^{-1}P_n(x)(T^*)^{-1}$, where $P_n(x) = \widetilde{A}_n^n x^n + \cdots$. These $\{P_n(x)\}_{n\geq 0}$ do

not satisfy the same recursion relation as ours. This choice of A_n^n above is motivated by the remarkable fact that then our $\{Q_n(x)\}_{n\geq 0}$ satisfy

$$Q_n(1)\mathbf{e}_d^* = \mathbf{e}_d^*, \tag{7.1}$$

where $\mathbf{e}_d$ is the d-dimensional row vector with all entries equal to 1. In other words, the sum of the elements in each row of $Q_n(1)$ gives the value 1.

Below, we give the explicit expression for the stochastic block tridiagonal matrix going with the sequence $\{Q_n(x)\}_{n\geq 0}$.

THEOREM 7.1. [illegible]

$$xQ_n(x) = A_nQ_{n+1}(x) + B_nQ_n(x) + C_nQ_{n-1}(x), \quad n \geq 0, \tag{7.2}$$

[illegible] $Q_{-1}(x) = 0$ [illegible] $Q_0(x) = I$ [illegible] $n \geq 0$, A_n [illegible]

$$A_n = \sum_{i=0}^{d-1} \frac{(k+n)(\beta+n+d)(\alpha+\beta+n+d+i)(\alpha+\beta-k+n+i+1)}{(k+n+d-i-1)(\alpha+\beta-k+n+2i+1)(\alpha+\beta+2n+d+i)_2} E_{ii}$$

$$+ \sum_{i=0}^{d-2} \frac{(i+1)(k+n)(k+d-i-2)(\beta+n+d)}{(\alpha+\beta+2n+d+i+1)(\alpha+\beta-k+n+2i+3)(k+n+d-i-2)_2} E_{i+1,i},$$

[illegible] $n \geq 1$, C_n [illegible]

$$C_n = \sum_{i=0}^{d-1} \frac{n(\alpha+n+i)(k+n+d-1)(\alpha+\beta-k+n+d+i)}{(k+n+d-i-1)(\alpha+\beta-k+n+2i+1)(\alpha+\beta+2n+d+i-1)_2} E_{ii}$$

$$+ \sum_{i=0}^{d-2} \frac{n(d-i-1)(k+n+d-1)(\beta-k+i+1)}{(\alpha+\beta+2n+d+i)(\alpha+\beta-k+n+2i+1)(k+n+d-i-2)_2} E_{i,i+1},$$

$n \geq 0$, B_n [illegible]

$$B_n = \sum_{i=0}^{d-1} \Bigg(1 + \frac{n(k+n-1)(k+n+d-1)(\beta+n+d-1)}{(\alpha+\beta+2n+d+i-1)(k+n+d-i-2)_2}$$

$$- \frac{(n+1)(k+n)(k+n+d)(\beta+n+d)}{(\alpha+\beta+2n+d+i+1)(k+n+d-i-1)_2}$$

$$+ \frac{i(d-i)(k+d-i-1)(\beta-k+i)}{(\alpha+\beta-k+n+2i)(k+n+d-i-1)_2}$$

$$- \frac{(i+1)(d-i-1)(k+d-i-2)(\beta-k+i+1)}{(\alpha+\beta-k+n+2i+2)(k+n+d-i-2)_2} \Bigg) E_{ii}$$

$$+ \sum_{i=0}^{d-2} \frac{(d-i-1)(\beta-k+i+1)(\alpha+\beta-k+n+i+1)(\alpha+\beta+n+d+i)}{(k+n+d-i-1)(\alpha+\beta+2n+d+i)(\alpha+\beta-k+n+2i+1)_2} E_{i,i+1}$$

$$+ \sum_{i=0}^{d-2} \frac{(i+1)(\alpha+n+i+1)(k+d-i-2)(\alpha+\beta-k+n+d+i+1)}{(k+n+d-i-2)(\alpha+\beta+2n+d+i+1)(\alpha+\beta-k+n+2i+2)_2} E_{i+1,i}.$$

[illegible]. The tools required to prove these formulas are implicitly included in [12] and [29] for the special case of $\beta = k = 1$. The detailed proof of these results can be obtained following the program described above. □

The advantage of dealing with this equivalent classical pair $\{\widetilde{W}, \widetilde{D}\}$, normalized as above, shows up, for instance, in the fact that we have a stochastic matrix. The entries of A_n, B_n, and C_n are nonnegative, and by applying both sides of the identity (7.2) to the vector $\mathbf{e}_d^*$, setting $x = 1$, and using (7.1) we obtain that the sum of the entries in each row of our block tridiagonal matrix equals one.

In [31] one finds a nice and different proof for the fact that our Jacobi matrix is stochastic in the special case of $\beta = k = 1$.

There are further advantages in dealing with the pair $\{\widetilde{W}, \widetilde{D}\}$: an expression for the norms of the family $\{Q_n(x)\}_{n\geq 0}$ with respect to the matrix measure $\widetilde{W}$, defined in (3.1), is given by the diagonal matrix

(7.3)

$$\|Q_n\|^2_{\widetilde{W}} = (Q_n, Q_n)_{\widetilde{W}} = \sum_{i=0}^{d-1} (-1)^i \frac{\Gamma(n+1)\Gamma(\beta+d)\Gamma(\alpha+n+i+1)(1-k-d-n)_i}{\binom{d-1}{i}\Gamma(d)\Gamma(\alpha+\beta+d+i+2n+1)}$$

$$\times \frac{(k+d-i-1)_n(\alpha+\beta+d+i+n)_n(\alpha+\beta-k+i+n+1)_d}{(\alpha+\beta-k+2i+n+1)(k)_n(\beta-k+1)_i(\beta+d)_n} E_{ii},$$

where Γ is the standard Gamma function. We point out that these matrix valued norms are given in our case by diagonal matrices, a fact that will play an important role later on.

For the benefit of the reader we include here the expression for all the quantities above in the case $d = 1$. The weight $\widetilde{W}$ and differential operator $\widetilde{D}$ are given by

$$\widetilde{W}(x) = x^\alpha (1-x)^\beta, \quad \widetilde{D} = x(1-x)\frac{d^2}{dx^2} + (\alpha + 1 + x(\alpha+\beta+2))\frac{d}{dx}.$$

Hence, we have that the coefficients of the three term recursion relation and the norms of the orthogonal polynomials are given by

$$A_n = \frac{(n+\beta+1)(n+\alpha+\beta+1)}{(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)}, \quad n \geq 0,$$

$$B_n = 1 + \frac{n(n+\beta)}{2n+\alpha+\beta} - \frac{(n+1)(n+\beta+1)}{2n+\alpha+\beta+2}, \quad n \geq 0,$$

$$C_n = \frac{n(n+\alpha)}{(2n+\alpha+\beta)(2n+\alpha+\beta+1)}, \quad n \geq 1,$$

$$\|Q_n\|^2_{\widetilde{W}} = \frac{\Gamma(n+1)\Gamma(n+\alpha+1)\Gamma(\beta+1)^2}{\Gamma(n+\beta+1)\Gamma(n+\alpha+\beta+1)(2n+\alpha+\beta+1)}, \quad n \geq 0.$$

The polynomials $\{Q_n(x)\}_{n\geq 0}$ are exactly the Jacobi polynomials in the interval $[0,1]$, normalized to satisfy $Q_n(1) = 1$.

The case $d = 2$, for the special case of $k = \frac{\beta+1}{2}$, can be found in [9].

8. Probabilistic aspects of our family of examples. The corresponding Jacobi matrix

$$P = \mathcal{L} = \begin{pmatrix} B_0 & A_0 & & & \\ C_1 & B_1 & A_1 & & \\ & C_2 & B_2 & A_2 & \\ & & \ddots & \ddots & \ddots \end{pmatrix} \tag{8.1}$$

made up from the coefficients introduced in Theorem 7.1 is stochastic; that is, $P\mathbf{e} = \mathbf{e}$, where $\mathbf{e}$ denotes the semi-infinite column vector with all entries equal to 1. Therefore, it gives a block tridiagonal transition probability matrix depending on three parameters, α, β, and k.

The Markov process that results from P is [illegible] and [illegible]. Indeed, one can see that for any pair of states (i,j), (i',j'), every entry in the (i,i')-block of P^n is positive if n is large enough.

THEOREM 8.1. [illegible] P [illegible] $-1 < \beta \leq 0$ [illegible] $\beta > 0$ [illegible].

. One can use directly the Karlin–McGregor representation formula (4.1) to obtain Corollary 4.1 of [3]. The process turns out to be recurrent if and only if

$$e_j^T \left(\int_0^1 \frac{\widetilde{W}(x)}{1-x} S_0^{-1} dx \right) e_j = \infty \tag{8.2}$$

for some $j \in \{1, \ldots, d\}$, where $e_j^T = (0, \ldots, 0, 1, 0, \ldots, 0)$ denotes the jth unit vector and $S_0 = \int_0^1 \widetilde{W}(x)dx = \|Q_0\|^2_{\widetilde{W}}$ is the first moment. Otherwise, the process is transient. The explicit expression of the weight matrix in our case is $\widetilde{W}(x) = x^\alpha (1-x)^\beta \widetilde{Z}(x)$, where $\widetilde{Z}(x)$ is a matrix polynomial and a detailed look shows that

$$\widetilde{Z}(1) = \frac{(\beta+1)_{d-1}}{(d-1)!} \sum_{i,j=0}^{d-1} E_{ij}.$$

Hence, condition (8.2) holds if and only if $-1 < \beta \leq 0$.

Corollary 4.2 of [3] gives a necessary and sufficient condition for a process to be positive recurrent. This happens exactly when one of the entries of the measure $\widetilde{W}$ has a jump at the point 1. But this is not true in our case given the form of $\widetilde{W}$ indicated above. So our process is never positive recurrent. This means that for $-1 < \beta \leq 0$ the process is null recurrent. □

All the considerations above result from explicit expressions for $\widetilde{W}(x)$ and the entries of P.

Now we come to a very delicate issue: the explicit computation of an invariant distribution. As noticed in (7.3), our matrix valued orthogonal polynomials $\{Q_n(x)\}_{n\geq 0}$ have the remarkable property that their norms are diagonal matrices. This provides us, for each n, with d scalars which could be used (inspired by the case of birth-and-death processes) as a way of getting an invariant row vector

$$\boldsymbol{\pi} = (\boldsymbol{\pi}^0; \boldsymbol{\pi}^1; \dots)$$

with the property that

$$\boldsymbol{\pi} P = \boldsymbol{\pi}.$$

Thus, we find the remarkable fact that the components of $\boldsymbol{\pi}$ can be computed by the recipe

$$\boldsymbol{\pi}^n = \mathbf{e}_d(\|Q_n\|^2_{\widetilde{W}})^{-1}, \quad n \geq 0,$$

where $\mathbf{e}_d$ is the d-dimensional row vector with all entries equal to 1. The fact that the process is never positive recurrent implies that there exists no invariant distribution such that $\boldsymbol{\pi}\mathbf{e} < \infty$.

The unicity of the invariant distribution $\boldsymbol{\pi}$ holds as a consequence of the extended Perron–Frobenius theorem for countable nonnegative matrices (see [32, Theorem 5.4]), when the process is recurrent, i.e., $-1 < \beta \leq 0$. However, we have extensive numerical evidence that this is true for all values of $\beta > -1$.

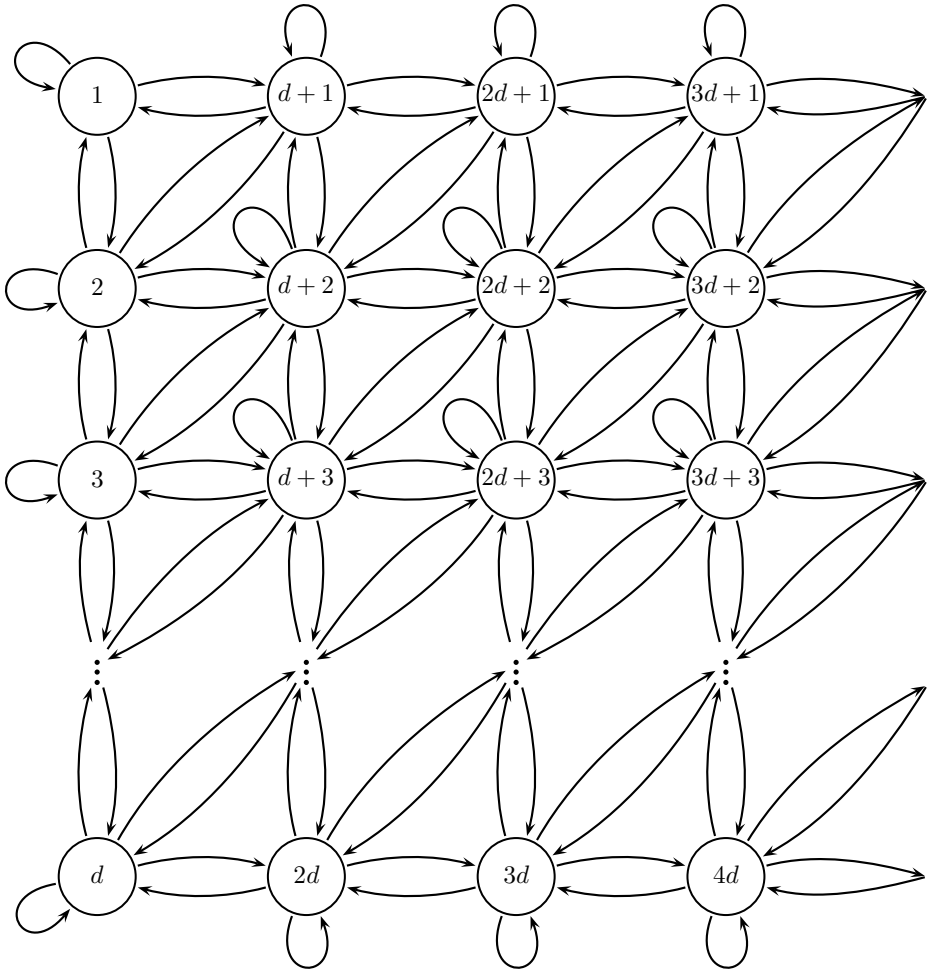
The case of random walk on the integers with general values of p and q ($p+q=1$) treated in [20] gives (for $p \neq q$) an example of a transient process where the invariant distribution is not unique. As mentioned above, it appears that even for values of β when our process is transient we still have a unique invariant distribution which can be computed explicitly in terms of the norms of our orthogonal polynomials.

To conclude this section we exhibit the network associated with our family of examples. The states of our network are labeled (as in any two dimensional situation) by two indices $i = 0, 1, 2, \dots$ and $j = 1, 2, \dots, d$. However, to write down a one-step transition probability matrix, one needs to agree on some linear order for the states. We use the convenient ordering

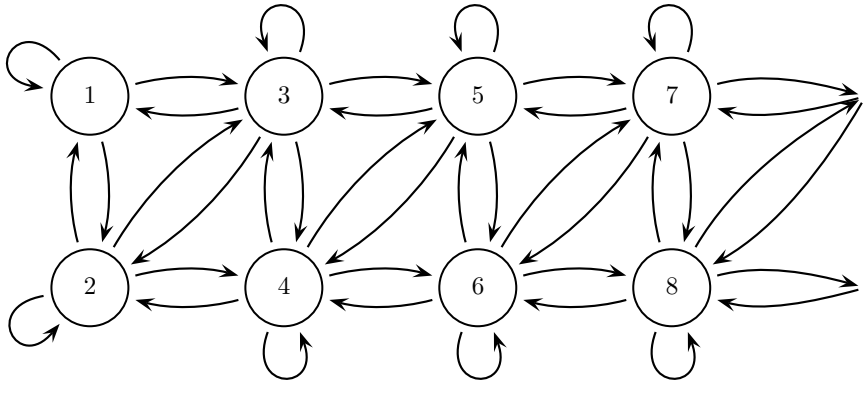
$$(0,1), (0,2), \dots, (0,d), (1,1), (1,2), \dots, (1,d), (2,1), (2,2), \dots, (2,d), \dots$$

so that, for instance, the label 3 in the following graph refers to $(0,3)$, while the label $d+2$ refers to $(1,2)$, etc. This choice of lexicographic order in one of the components and then in the other is an unpleasant feature that cannot be avoided completely.

The state space and the corresponding one-step transitions appear as follows:



9. The shape of the invariant distribution. In this section we will study in more detail the behavior of the invariant distribution when the number of phases d is equal to two, a luxury we can afford since we have an analytic expression. In this case, the associated network takes the form



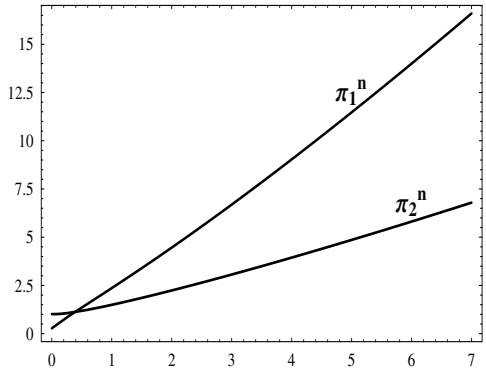


FIG. 9.1. $\alpha = -0.9, \beta = 0.1, k = 0.8$.

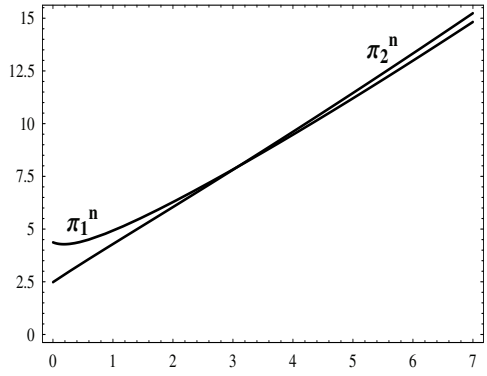


FIG. 9.2. $\alpha = 2.5, \beta = 0.1, k = 0.55$.

The invariant distribution $\boldsymbol{\pi}$ such that $\boldsymbol{\pi} P = \boldsymbol{\pi}$ is given by

$$\boldsymbol{\pi} = (\boldsymbol{\pi}^0; \boldsymbol{\pi}^1; \dots),$$

where $\boldsymbol{\pi}^n, n \geq 0$, is the 2-dimensional vector given by

$$\boldsymbol{\pi}^n = \frac{\Gamma(n+\alpha+\beta+2)\Gamma(n+\beta+2)}{\Gamma(n+\alpha+1)\Gamma(n+1)\Gamma(\beta+2)^2(n+\alpha+\beta-k+2)} \left(\frac{k(2n+\alpha+\beta+2)}{n+k} , \frac{(\beta-k+1)(2n+\alpha+\beta+3)(n+\alpha+\beta+2)}{(n+\alpha+1)(n+k+1)} \right).$$

From this explicit expression we can easily obtain several quantities. Data of special interest may be the initial value and the asymptotic behavior. The initial value is given by

$$\boldsymbol{\pi}^0 = \frac{\Gamma(\alpha+\beta+3)}{\Gamma(\alpha+1)\Gamma(\beta+2)(\alpha+\beta-k+2)} \left(1 , \frac{(\beta-k+1)(\alpha+\beta+3)}{(\alpha+1)(k+1)} \right).$$

The asymptotic behavior follows using asymptotic formulas for the Gamma function such as $\frac{\Gamma(z+\alpha)}{\Gamma(z)} \approx z^\alpha$ as $|z| \to \infty$. Hence, we have

$$\lim_{n\to\infty} \boldsymbol{\pi}^n = \begin{cases} (\infty, \infty) & \text{if} \quad \beta > -\frac{1}{2}, \\ \frac{4}{\pi}(2k, 1-2k) & \text{if} \quad \beta = -\frac{1}{2}, \\ (0,0) & \text{if} \quad -1 < \beta < -\frac{1}{2}. \end{cases}$$

In what follows we shall include plots of the two components $\boldsymbol{\pi}_1^n$ and $\boldsymbol{\pi}_2^n$, as functions of n, in a few representative cases. The general shape of both curves can change depending on the values of the parameters α, β, and k. A look at the role of the parameter β gives rise to four regions, namely $-1 < \beta < -1/2$, $\beta = -1/2$, $-1/2 < \beta < 0$, and $\beta \geq 0$. The parameter α only has influence on cosmetic changes like curvature and initial values depending on $-1 < \alpha < 0$ or $\alpha \geq 0$, while k affects the shape of the plots when its value is the middle point $\frac{\beta+1}{2}$ of its possible range and the situation in the rest of values is quite symmetric.

Figures 9.1 and 9.2 show the most interesting situations when $\beta > 0$.

Figures 9.3 and 9.4 show how the situation can change for small perturbations around $\beta = -1/2$. In Figure 9.3 both curves have a logarithmic growth and the second component has a minimum, while in Figure 9.4 both curves tend to 0.

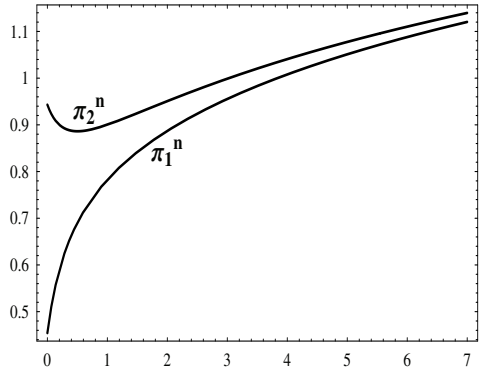


FIG. 9.3. $\alpha = -0.8, \beta = -0.4, k = 0.3$.

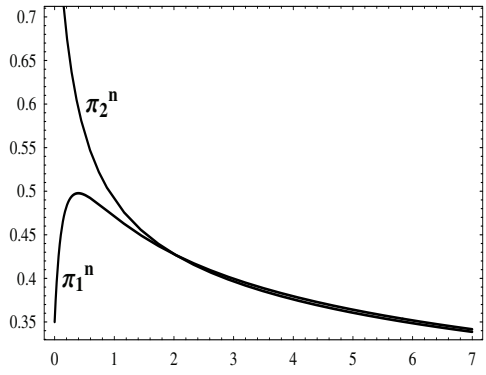


FIG. 9.4. $\alpha = -0.9, \beta = -0.6, k = 0.2$.

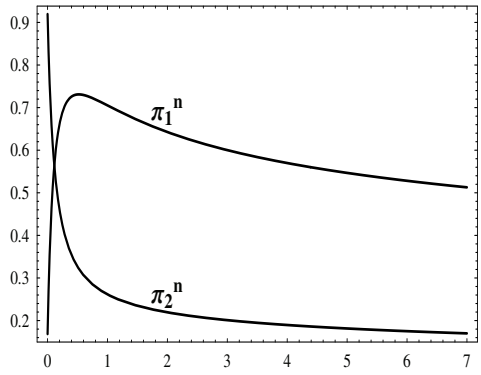


FIG. 9.5. $\alpha = -0.98, \beta = -0.6, k = 0.3$.

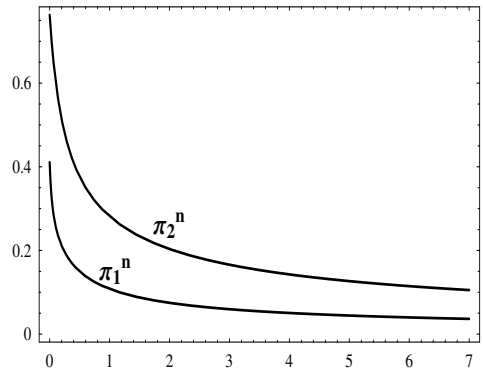


FIG. 9.6. $\alpha = -0.9, \beta = -0.8, k = 0.05$.

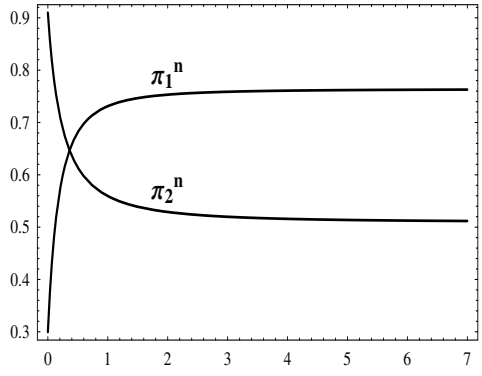


FIG. 9.7. $\alpha = -0.92, \beta = -0.5, k = 0.3$.

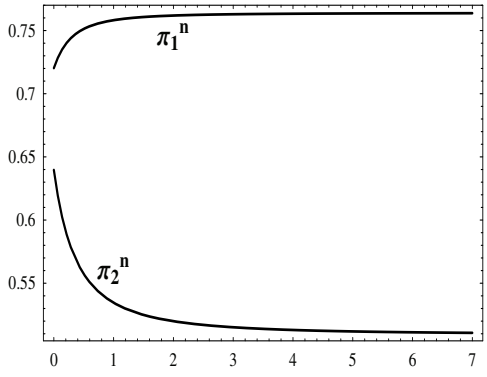


FIG. 9.8. $\alpha = -0.6857, \beta = -0.5, k = 0.3$.

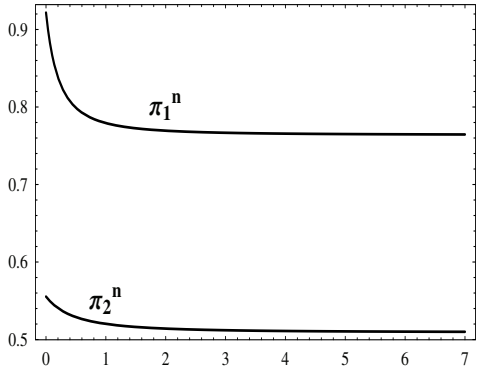


FIG. 9.9. $\alpha = -0.4857, \beta = -0.5, k = 0.3$.

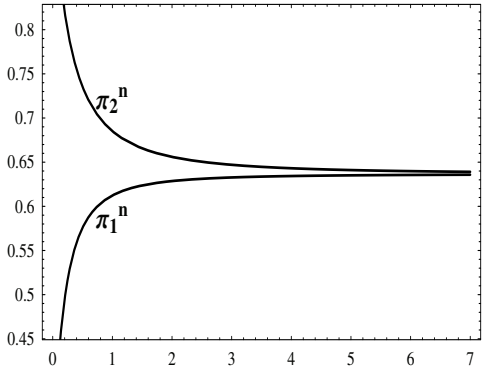


FIG. 9.10. $\alpha = -0.9, \beta = -0.5, k = 0.25$.

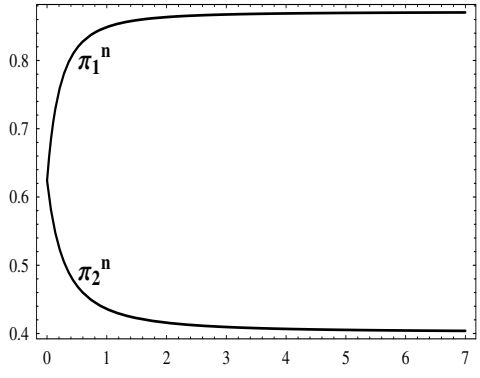


FIG. 9.11. $\alpha = -0.8, \beta = -0.5, k = 0.3421$.

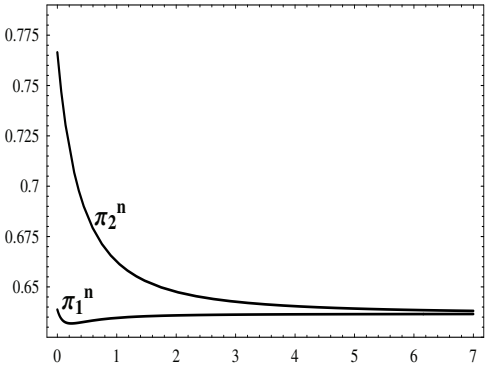


FIG. 9.12. $\alpha = -0.7, \beta = -0.5, k = 0.25$.

We observe that Figure 9.5, with k now approaching $\beta+1$, is similar to Figure 9.4. In Figure 9.6 we observe the consequences of k being very small.

The remaining figures refer to the case $\beta = -1/2$, where both components converge for large n. In Figure 9.7 we observe that the initial value of the first component is always lower than that of the second component and that the initial value of the second component is always greater than that of the first component. A small change of the value of α has the effect that the first component is always greater than the second component, as we can see in Figure 9.8.

Small perturbations on α change the curvatures of the components in Figure 9.9 with respect to Figure 9.8. In Figure 9.10 both components tend to the same value without ever touching, a consequence of choosing $k = \frac{\beta+1}{2}$.

The last two figures show how the situation can change for small perturbations of α and k. In Figure 9.11 both curves start from the same value and then they converge to different limits, while in Figure 9.12 both components converge to the same limit.

10. Concluding remarks. A block tridiagonal matrix $\mathcal{L}$ with nonnegative entries and individual rows that add up to 1 gives rise to a quasi-birth-and-death process. The explicit evaluation of $\mathcal{L}^n$, for arbitrary $n = 1, 2, 3, \ldots$, can be greatly simplified by using ideas that go back to Karlin and McGregor and have been explicitly set forth in [3, 10]. The only major difficulty here is that of computing the weight matrix $W(x)$. In this paper we start from a rich group theoretical situation that yields $W(x)$ as well

as a matrix $\mathcal{L}$ of the type envisaged previously. There are enough free parameters here to give instances of transient as well as recurrent Markov chains. It remains as interesting challenge to find some real life applications to this large collection of examples.

REFERENCES

[1] S. Basu and N. K. Bose, *Matrix Stieltjes series and network models*, SIAM J. Matrix Anal. Appl., 14 (1983), pp. 209–222.

[2] Ju. M. Berezans'kiĭ, *Expansions in Eigenfunctions of Selfadjoint Operators*, Transl. Math. Monogr. 17, American Mathematical Society, Providence, RI, 1968.

[3] H. Dette, B. Reuther, W. J. Studden, and M. Zygmunt, *Matrix measures and random walks with a block tridiagonal transition matrix*, SIAM J. Matrix Anal. Appl., 29 (2006), pp. 117–142.

[4] A. J. Durán, *On orthogonal polynomials with respect to positive definite matrix of measures*, Canad. J. Math., 47 (1995), pp. 88–112.

[5] A. J. Durán, *Matrix inner product having a matrix symmetric second order differential operator*, Rocky Mountain J. Math., 27 (1997), pp. 585–600.

[6] A. J. Durán and F. A. Grünbaum, *Orthogonal matrix polynomials satisfying second order differential equations*, Int. Math. Res. Not., 10 (2004), pp. 461–484.

[7] W. Feller, *On second order differential operators*, Ann. of Math. (2), 61 (1955), pp. 90–105.

[8] I. J. Good, *Random motion and analytic continued fractions*, Math. Proc. Cambridge Philos. Soc., 54 (1958), pp. 43–47.

[9] F. A. Grünbaum, *Matrix valued Jacobi polynomials*, Bull. Sci. Math., 127 (2003), pp. 207–214.

[10] F. A. Grünbaum, *Random walks and orthogonal polynomials: Some challenges*, Probability, Geometry and Integrable Systems, Vol. 55, MSRI Publications, Cambridge University Press, Cambridge, UK, pp. 241–260.

[11] F. A. Grünbaum and M. D. de la Iglesia, *Matrix valued orthogonal polynomials related to $SU(N+1)$, their algebras of differential operators and the corresponding curves*, Experiment. Math., 16 (2007), pp. 189–207.

[12] F. A. Grünbaum, I. Pacharoni, and J. A. Tirao, *A matrix-valued solution to Bochner's problem*, J. Phys. A: Math. Gen., 34 (2001), pp. 10647–10656.

[13] F. A. Grünbaum, I. Pacharoni, and J. A. Tirao, *Matrix valued spherical functions associated to the complex projective plane*, J. Funct. Anal., 188 (2002), pp. 350–441.

[14] F. A. Grünbaum, I. Pacharoni, and J. A. Tirao, *Matrix valued orthogonal polynomials of the Jacobi type*, Indag. Math. (N.S.), 14 (2003), pp. 353–366.

[15] F. A. Grünbaum, I. Pacharoni, and J. A. Tirao, *Matrix valued orthogonal polynomials of Jacobi type: The role of group representation theory*, Ann. Inst. Fourier (Grenoble), 55 (2005), pp. 2051–2068.

[16] F. A. Grünbaum, I. Pacharoni, and J. A. Tirao, *An invitation to matrix valued spherical functions: Linearization of products in the case of the complex projective space $P_2(\mathbb{C})$*, in Modern Signal Processing, D. Healy and D. Rockmore, eds., Cambridge University Press, Cambridge, UK, 2004, pp. 147–160.

[17] T. E. Harris, *First passages and recurrence distributions*, Trans. Amer. Math. Soc., 73 (1952), pp. 471–486.

[18] M. E. H. Ismail, J. Letessier, D. Masson, and G. Valent, *Birth and death processes and orthogonal polynomials*, in Orthogonal Polynomials, P. Nevai, ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990, pp. 229–255.

[19] M. Kac, *Random walk and the theory of Brownian motion*, Amer. Math. Monthly, 54 (1947), pp. 369–391.

[20] S. Karlin and J. McGregor, *Random walks*, IIlinois J. Math., 3 (1959), pp. 66–81.

[21] M. G. Kreĭn, *Fundamental aspects of the representation theory of Hermitian operators with deficiency index (m,m)*, Amer. Math. Soc. Transl. Ser. 2, 97 (1971), Providence, RI, pp. 75–143.

[22] M. G. Kreĭn, *Infinite J-matrices and a matrix moment problem*, Dokl. Akad. Nauk SSSR, 69 (1949), pp. 125–128.

[23] G. Latouche, C. E. M. Pearce, and P. G. Taylor, *Invariant measures for quasi-birth-and-death processes*, Comm. Statist. Stochastic Models, 14 (1998), pp. 443–460.

[24] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*, ASA-SIAM Ser. Stat. Appl. Probab. 5, SIAM, Philadelphia, 1999.

[25] W. Ledermann and G. E. Reuter, *Spectral theory for the differential equations of simple birth and death processes*, Philos. Trans. Roy. Soc. London, Ser. A., 246 (1954), pp. 321–369.
[26] H. P. McKean, Jr., *Elementary solutions for certain parabolic partial differential equations*, Trans. Amer. Math. Soc., 82 (1956), pp. 519–548.
[27] M. F. Neuts, *Structured Stochastic Matrices of $M/G/1$ Type and Their Applications*, Marcel Dekker, New York, 1989.
[28] I. Pacharoni and P. Román, *A sequence of matrix valued orthogonal polynomials associated to spherical functions*, Constr. Approx., 28 (2007), pp. 127–147.
[29] I. Pacharoni and J. A. Tirao, *Three term recursion relation for spherical functions associated to the complex projective plane*, Math. Phys. Anal. Geom., 7 (2004), pp. 193–221.
[30] I. Pacharoni and J. A. Tirao, *Matrix valued orthogonal polynomials arising from the complex projective space*, Constr. Approx., 25 (2007), pp. 177–192.
[31] I. Pacharoni and J. A. Tirao, *Three term recursion relation for spherical functions associated to the complex hyperbolic plane*, J. Lie Theory, 17 (2007), pp. 791–828.
[32] E. Seneta, *Non-negative Matrices and Markov Chains*, 3rd ed., Springer-Verlag, New York, 2006.
[33] J. Shohat and J. Tamarkin, *The Problem of Moments*, American Mathematical Society Mathematical Surveys II, American Mathematical Society, Providence, RI, 1943.
[34] J. A. Tirao, *Spherical functions*, Rev. de la Unión Matem. Argentina, 28 (1977), pp. 75–98.
[35] J. A. Tirao, *The matrix valued hypergeometric equation*, Proc. Nat. Acad. Sci. USA, 100 (2003), pp. 8138–8141.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 762–776

PERTURBATION BOUNDS FOR DETERMINANTS AND CHARACTERISTIC POLYNOMIALS*

ILSE C. F. IPSEN† AND RIZWANA REHMAN†

Abstract. We derive absolute perturbation bounds for the coefficients of the characteristic polynomial of a $n \times n$ complex matrix. The bounds consist of elementary symmetric functions of singular values, and suggest that coefficients of normal matrices are better conditioned with regard to absolute perturbations than those of general matrices. When the matrix is Hermitian positive-definite, the bounds can be expressed in terms of the coefficients themselves. We also improve absolute and relative perturbation bounds for determinants. The basis for all bounds is an expansion of the determinant of a perturbed diagonal matrix.

Key words. elementary symmetric functions, singular values, eigenvalues, condition number, determinant

AMS subject classifications. 65F40, 65F15, 65F35, 65Z05, 15A15, 15A18

DOI. 10.1137/070704770

1. Introduction.

The characteristic polynomial of a $n \times n$ complex matrix A is defined as

$$\det(\lambda I - A) = \lambda^n + c_1\lambda^{n-1} + \cdots + c_{n-1}\lambda + c_n,$$

where in particular $c_n = (-1)^n \det(A)$ and $c_1 = -\operatorname{trace}(A)$.

The coefficients of the characteristic polynomial of a complex matrix are of central importance in a quantum physics application. There they supply information about thermodynamic properties of fermionic systems, which arise, for instance, in the study of structure and evolution of neutron stars. These thermodynamic quantities are calculated from partition functions. It turns out that the partition function Z_k for k noninteracting fermions is given by $Z_k = (-1)^k c_k$, where the matrix A is a function of the particle Hamiltonian operator [9]. Partition functions for systems of interacting fermions require repeated calculation of noninteracting partition functions. The matrices A in these problems have fairly small dimension ($n \leq 1000$) and no discernible structure.

In order to assess the stability of numerical methods for computing the characteristic polynomial, though, we first need to know the conditioning of the c_k and their sensitivity to perturbations in the matrix A. To this end, we derive perturbation bounds for absolute normwise perturbations.

Main results. The main idea behind our perturbation bounds is a determinant expansion of a perturbed diagonal matrix (Theorem 2.3). The expansion can be extended to any square matrix via the SVD (Corollary 2.4). The resulting absolute perturbation bounds contain elementary symmetric functions of singular values. Below we present weaker, first-order versions of these bounds.

Let $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ be the singular values of A, and let $A + E$ be a $n \times n$ complex matrix with $\|E\|_2 < 1$ and characteristic polynomial $\det(\lambda I - (A + E)) =$

*Received by the editors October 8, 2007; accepted for publication (in revised form) by A. Frommer April 3, 2008; published electronically July 2, 2008.
http://www.siam.org/journals/simax/30-2/70477.html

†Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205 (ipsen@ncsu.edu, http://www4.ncsu.edu/~ipsen/, rrehman@unity.ncsu.edu).

$\lambda^n + \tilde{c}_1\lambda^{n-1} + \cdots + \tilde{c}_{n-1}\lambda + \tilde{c}_n$. The extreme coefficients c_1 and c_n have the simplest bounds. The linearity of the trace implies

$$|\tilde{c}_1 - c_1| = |\operatorname{trace}(A+E) - \operatorname{trace}(A)| = |\operatorname{trace}(E)| \leq n\|E\|_2,$$

so that coefficient c_1 is well conditioned with regard to absolute perturbations if the matrix order n is not too large. The determinant satisfies to first order (Remark 2.9)

$$|\tilde{c}_n - c_n| = |\det(A) - \det(A+E)| \leq s_{n-1}\|E\|_2 + \mathcal{O}(\|E\|_2^2),$$

where s_{n-1} is the $(n-1)$st elementary symmetric function in the singular values and has the upper bound $s_{n-1} \leq n\sigma_1 \ldots \sigma_{n-1}$.

The remaining coefficients c_k satisfy to first order (Remark 3.4)

$$|\tilde{c}_k - c_k| \leq \binom{n}{k} s_{k-1}^{(k)}\|E\|_2 + \mathcal{O}(\|E\|_2^2), \qquad 2 \leq k \leq n,$$

where $s_{k-1}^{(k)}$ is the $(k-1)$st elementary symmetric function in the k largest singular values, and $s_{k-1}^{(k)} \leq k\sigma_1 \ldots \sigma_{k-1}$. However, if the matrix is normal or Hermitian, then the bound improves to (Remark 3.6)

$$|\tilde{c}_k - c_k| \leq (n-k+1)s_{k-1}\|E\|_2 + \mathcal{O}(\|E\|_2^2), \qquad 2 \leq k \leq n,$$

where s_{k-1} is the $(k-1)$st function in all singular values. Also, since A is normal, $\sigma_i = |\lambda_i|$, where λ_i are the eigenvalues. Since the binomial term $\binom{n}{k}$ is reduced to $n-k+1$, the coefficients of a normal matrix are likely to be better conditioned than those of a general matrix.

When the matrix is Hermitian positive-definite, eigenvalues are equal to singular values, and the above bound can be written as (Corollary 3.7)

$$|\tilde{c}_k - c_k| \leq (n-k+1)|c_{k-1}|\|E\|_2 + \mathcal{O}(\|E\|_2^2), \qquad 2 \leq k \leq n.$$

As a result, c_k is well conditioned in the absolute sense if the magnitude of the preceding coefficient $|c_{k-1}|$ is not too large.

Overview. Section 2 deals with determinants. We first derive expansions for determinants (section 2.1), and from them absolute perturbation bounds in terms of elementary symmetric functions of singular values (section 2.2), as well as relative bounds for determinants (section 2.3), and local sensitivity results (section 2.4). Section 3 deals with coefficients c_k of the characteristic polynomial. We derive absolute perturbation bounds for general matrices (section 3.1) and normal matrices (section 3.2), as well as normwise bounds (section 3.3).

Notation. The matrix A is a $n \times n$ complex matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$, and eigenvalues λ_i, labelled so that $|\lambda_1| \geq \cdots \geq |\lambda_n|$. The two-norm is $\|A\|_2 = \sigma_1$, and A^* is the conjugate transpose of A. The matrix $I = \operatorname{diag}\begin{pmatrix}1 & \ldots & 1\end{pmatrix}$ is the identity matrix, with columns e_i, $i \geq 1$. We denote by A_i the principal submatrix of order $n-1$ that is obtained by removing row and column i of A, and by $A_{i_1 \ldots i_k}$ the principal submatrix of order $n-k$, obtained by removing rows and columns $i_1 \ldots i_k$.

2. Determinants. We derive expansions and perturbation bounds for determinants. We start with expansions for determinants of perturbed matrices (section 2.1), and from them derive absolute perturbation bounds in terms of elementary symmetric functions of singular values (section 2.2), as well as relative bounds for determinants (section 2.3), and local sensitivity results (section 2.4).

2.1. Expansions. We derive expansions for determinants of perturbed matrices in several steps, by considering perturbations that have only a single nonzero diagonal element (Lemma 2.1), perturbations of diagonal matrices (Theorem 2.3), and at last perturbations of general matrices (Corollary 2.4).

LEMMA 2.1. [illegible] A [illegible] $n \times n$ [illegible] α [illegible] A_i [illegible] $n-1$ [illegible] i [illegible] A [illegible] $B = A + \alpha e_i e_i^*$ [illegible] $\det(B) = \det(A) + \alpha \det(A_i)$, $1 \le i \le n$ [illegible]. This follows from a cofactor expansion [8, Theorem 2.3.1] along row i or column i of B. ∎

The above expansion can be used to expand the determinant of a perturbed diagonal matrix. Before deriving this expansion, we motivate its expression on matrices of order 2 and 3.

[illegible] 2.2. If

$$D = \begin{pmatrix} \delta_1 & \\ & \delta_2 \end{pmatrix}, \qquad F = \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix},$$

then $\det(D+F) = \det(D) + \det(F) + S_1$, where $S_1 \equiv \delta_1 f_{22} + \delta_2 f_{11}$.

If

$$D = \begin{pmatrix} \delta_1 & & \\ & \delta_2 & \\ & & \delta_3 \end{pmatrix}, \qquad F = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix},$$

then $\det(D+F) = \det(D) + \det(F) + S_1 + S_2$, where

$$S_1 \equiv \delta_1 \det \begin{pmatrix} f_{22} & f_{23} \\ f_{32} & f_{33} \end{pmatrix} + \delta_2 \det \begin{pmatrix} f_{11} & f_{13} \\ f_{31} & f_{33} \end{pmatrix} + \delta_3 \det \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix},$$

and $S_2 \equiv \delta_1\delta_2 f_{33} + \delta_1\delta_3 f_{22} + \delta_2\delta_3 f_{11}$.

These examples illustrate that the expansion of $\det(D+F)$ can be written as a sum, where each term consists of a product of k diagonal elements of D and the determinant of the "complementary" submatrix of order $n-k$ of F.

To derive expansions for diagonal matrices of any order, we denote by $F_{i_1 \dots i_k}$ the principal submatrix of order $n-k$ obtained by deleting rows and columns $i_1 \dots i_k$ of the $n \times n$ matrix F.

THEOREM 2.3 (expansion for diagonal matrices). [illegible] D [illegible] F [illegible] $n \times n$ [illegible] $D = \operatorname{diag}(\delta_1 \ \dots \ \delta_n)$ [illegible]

$$\det(D+F) = \det(D) + \det(F) + S_1 + \cdots + S_{n-1},$$

[illegible]

$$S_k \equiv \sum_{1 \le i_1 < \cdots < i_k \le n} \delta_{i_1} \cdots \delta_{i_k} \det(F_{i_1 \dots i_k}), \qquad 1 \le k \le n-1.$$

[illegible] $\delta_1 = \cdots = \delta_j = 0$ [illegible] $1 \le j \le n-1$ [illegible]

$$\det(D+F) = \det(F) + S_1 + \cdots + S_{n-j},$$

[illegible]

$$S_k = \sum_{j+1 \le i_1 < \cdots < i_k \le n} \delta_{i_1} \cdots \delta_{i_k} \det(F_{i_1 \dots i_k}), \qquad 1 \le k \le n-j.$$

The proof is by induction over the matrix order n, and Example 2.2 represents the induction basis. Assuming the statement is true for matrices of order $n-1$, we show that it is also true for matrices of order n. Let

$$D^{(j)} \equiv \operatorname{diag}\begin{pmatrix} 0 & \dots & 0 & \delta_{j+1} & \dots & \delta_n \end{pmatrix}$$

be a diagonal matrix of order n with j leading zeros. Applying Lemma 2.1 to $A \equiv D^{(1)} + F$ and $B \equiv A + \delta_1 e_1 e_1^*$ gives

$$\det(D + F) = \delta_1 \det(D_1 + F_1) + \det(D^{(1)} + F).$$

We repeat this process on the second summand $\det(D^{(1)} + F)$ to remove the diagonal elements δ_j one by one; $j \geq 2$. To this end, we apply Lemma 2.1 to $A \equiv D^{(j)} + F$ and $B \equiv A + \delta_j e_j e_j^*$, and denote by $(D^{(j)})_j$ the matrix of order $n-1$ obtained by removing row and column j from $D^{(j)}$. This gives

$$\det(D^{(1)} + F) = \sum_{j=2}^{n-1} \delta_j \det\left((D^{(j)})_j + F_j\right) + \delta_n \det(F_n) + \det(F).$$

Putting the above expression into the expansion for $\det(D + F)$ yields

$$\det(D + F) = \det(F) + \delta_1 \det(D_1 + F_1) + \sum_{j=2}^{n-1} \delta_j \det\left((D^{(j)})_j + F_j\right) + \delta_n \det(F_n).$$

Since $D_1 + F_1$ and $(D^{(j)})_j + F_j$ are matrices of order $n-1$, we can apply the induction hypothesis. To take advantage of the fact that the $j-1$ top diagonal elements of $(D^{(j)})_j$ are zero, we define the following sums for matrices of order $n-1$,

$$S_k^{(j)} \equiv \sum_{j+1 \leq i_1 < \dots < i_k \leq n} \delta_{i_1} \cdots \delta_{i_k} \det(F_{j i_1 \dots i_k}), \qquad 1 \leq j \leq n-1, \quad 1 \leq k \leq n-j,$$

where $F_{j i_1 \dots i_k}$ is the matrix of order $n-k-1$ obtained by removing rows and columns $j, i_1, \dots, i_k$ of F. The induction hypothesis yields

$$\begin{aligned} \det(D_1 + F_1) &= \det(D_1) + \det(F_1) + S_1^{(1)} + \dots + S_{n-2}^{(1)}, \\ \det\left((D^{(j)})_j + F_j\right) &= \det(F_j) + S_1^{(j)} + \dots + S_{n-j}^{(j)}, \qquad 2 \leq j \leq n-2, \\ \det\left((D^{(n-1)})_{n-1} + F_{n-1}\right) &= \det(F_{n-1}) + S_1^{(n-1)}. \end{aligned}$$

Now substitute the above expansions into the expression for $\det(D + F)$ and use the fact that $\delta_1 \det(D_1) = \det(D)$, $\sum_{i=1}^{n} \delta_i \det(F_i) = S_1$, and

$$\sum_{i=1}^{n-j} \delta_i S_j^{(i)} = S_{j+1}, \qquad 1 \leq j \leq n-2. \qquad \square$$

When the leading j diagonal elements of D are zero, then at most $n-j$ of the S_k are nonzero, and within each S_k one needs to account only for the nonzero summands. We now extend Theorem 2.3 to general matrices, by transforming them to diagonal form via the SVD. Let $A = U \Sigma V^*$ be a SVD of A, where $\Sigma = \operatorname{diag}\begin{pmatrix} \sigma_1 & \dots & \sigma_n \end{pmatrix}$ with $\sigma_1 \geq \dots \geq \sigma_n \geq 0$, and U and V are unitary.

COROLLARY 2.4 (expansion for general matrices). Let A and E be $n \times n$ matrices, and let $F \equiv U^*EV$.

$$\det(A+E) = \det(A) + \det(E) + S_1 + \cdots + S_{n-1},$$

where

$$S_k \equiv \det(UV^*) \sum_{1 \le i_1 < \cdots < i_k \le n} \sigma_{i_1} \cdots \sigma_{i_k} \det(F_{i_1 \ldots i_k}), \qquad 1 \le k \le n-1.$$

If $\operatorname{rank}(A) = r$, $1 \le r \le n-1$, then

$$\det(A+E) = \det(E) + S_1 + \cdots + S_r,$$

where

$$S_k = \det(UV^*) \sum_{1 \le i_1 < \cdots < i_k \le r} \sigma_{i_1} \cdots \sigma_{i_k} \det(F_{i_1 \ldots i_k}), \qquad 1 \le k \le r.$$

Proof. The SVD of A implies $A + E = U(\Sigma + F)V^*$, and Theorem 2.3 implies

$$\det(\Sigma + F) = \det(\Sigma) + \det(F) + \hat{S}_1 + \cdots + \hat{S}_{n-1},$$

where

$$\hat{S}_k \equiv \sum_{1 \le i_1 < \cdots < i_k \le n} \sigma_{i_1} \cdots \sigma_{i_k} \det(F_{i_1,\ldots,i_k}), \qquad 1 \le k \le n-1.$$

With $S_k \equiv \det(UV^*)\hat{S}_k$ we obtain $\det(A+E) = \det(A) + \det(E) + S_1 + \cdots + S_{n-1}$.

Now suppose $\operatorname{rank}(A) = r \le n-1$. Then $n-r$ singular values are zero, so that all products of $r+1$ or more singular values are zero. In particular, $\det(A) = 0$. If $\operatorname{rank}(A) = r < n-1$, then $S_{r+1} = \cdots = S_{n-1} = 0$. Moreover, the terms $S_1, \ldots, S_r$ contain only the nonzero singular values $\sigma_1, \ldots, \sigma_r$. ∎

Corollary 2.4 shows that the number of summands in the expansion decreases with the rank of the matrix.

2.2. Absolute perturbation bounds. We derive absolute perturbation bounds for determinants in terms of elementary symmetric functions of singular values. These bounds give rise to absolute first-order condition numbers. We also derive simpler, but weaker normwise bounds.

To bound the perturbations we need the following inequalities.

LEMMA 2.5 (Hadamard's inequality). Let B be $n \times n$, and let $\|Be_i\|_2$ be the 2-norm of column i of B. Then

$$|\det(B)| \le \prod_{i=1}^{n} \|Be_i\|_2 \le \|B\|_2^n.$$

Proof. The first inequality is Hadamard's inequality [6, Corollary 7.8.2]. ∎

The bounds also contain elementary symmetric functions, which are defined as follows [6, Definition 1.2.9].

DEFINITION 2.6 (elementary symmetric functions of singular values). Let A be $n \times n$, and let $\sigma_1 \ge \ldots \ge \sigma_n$ be its singular values. Then

$$s_0 \equiv 1, \qquad s_k \equiv \sum_{1 \le i_1 < \cdots < i_k \le n} \sigma_{i_1} \cdots \sigma_{i_k}, \qquad 1 \le k \le n,$$

are the kth elementary symmetric functions of the singular values of A.

Now we are ready to derive the first perturbation bound for determinants of general matrices.

COROLLARY 2.7 (general matrices). [illegible] A [illegible] E [illegible] $n \times n$ [illegible]

$$|\det(A) - \det(A+E)| \le \sum_{i=1}^{n} s_{n-i}\|E\|_2^i.$$

[illegible] $\operatorname{rank}(A) = r$ [illegible] $1 \le r \le n-1$ [illegible]

$$|\det(A+E)| \le \|E\|_2^{n-r} \sum_{i=0}^{r} s_{r-i}\|E\|_2^i,$$

[illegible] s_j [illegible] r [illegible] A, $1 \le j \le r$

[illegible] $E = \epsilon UV^*$ [illegible] $\epsilon > 0$, [illegible] $A = U\Sigma V^*$ [illegible] A

[illegible]. Corollary 2.4 implies $|\det(A) - \det(A+E)| \le |\det(E)| + |S_1| + \cdots + |S_{n-1}|$. To bound $|S_k|$ use the fact that $|\det(UV^*)| = 1$ and $\sigma_i \ge 0$ to obtain

$$\begin{aligned} |S_k| &\le \max_{1 \le i_1 < \cdots < i_k \le n} |\det(F_{i_1 \ldots i_k})| \sum_{1 \le i_1 < \cdots < i_k \le n} \sigma_{i_1} \cdots \sigma_{i_k} \\ &= \max_{1 \le i_1 < \cdots < i_k \le n} |\det(F_{i_1 \ldots i_k})| s_k. \end{aligned}$$

Lemma 2.5 implies $|\det(E)| \le \|E\|_2^n$, and $|\det(F_{i_1 \ldots i_k})| \le \|F\|_2^{n-k} = \|E\|_2^{n-k}$. Hence $|S_k| \le s_k \|E\|_2^{n-k}$, $1 \le k \le n-1$.

Now suppose $\operatorname{rank}(A) = r$. Then Corollary 2.4 implies

$$|\det(A+E)| \le |\det(E)| + |S_1| + \cdots + |S_r| \le \|E\|_2^{n-r} \sum_{i=0}^{r} s_{r-i}\|E\|_2^i,$$

where the terms s_{r-i} contain only nonzero singular values.

If $E = \epsilon UV^*$, then $F = \epsilon I$ and $\det(F_{i_1 \ldots i_k}) = \epsilon^{n-k} = \|E\|_2^{n-k}$, so that $S_k = |S_k| = \|E\|_2^{n-k} s_k$. □

Corollary 2.7 bounds the absolute error in $\det(A+E)$ by elementary symmetric functions of singular values and powers of $\|E\|_2$. Although the bounds for nonsingular and rank-r matrices look different, because the sums start at different indices, they are consistent. If $\operatorname{rank}(A) \le n-k$ for some $k \ge 1$, then $|\det(A+E)|$ is bounded by a multiple of $\|E\|_2^k$. Hence if $\|E\|_2 < 1$ then determinants of rank-deficient matrices tend to be better conditioned in the absolute sense.

[illegible] 2.8 (Hermitian positive-definite matrices). In the special case when A is Hermitian positive-definite, singular values are equal to eigenvalues, so that we can write the elementary symmetric functions in terms of the eigenvalues $\lambda_1 \ge \cdots \ge \lambda_n \ge 0$. Hence in Corollary 2.7

$$s_k = \sum_{1 \le i_1 < \cdots < i_k \le n} \lambda_{i_1} \cdots \lambda_{i_k}, \qquad 1 \le k \le n-1.$$

Note that $A+E$ does not have to be Hermitian positive-definite, because no restrictions are placed on E.

[illegible] 2.9 (first-order absolute condition numbers). Let A be a $n \times n$ complex matrix with $\operatorname{rank}(A) \ge n-1$ and $\|E\|_2 < 1$. Corollary 2.7 implies the first-order bound

$$|\det(A) - \det(A+E)| \le s_{n-1}\|E\|_2 + \mathcal{O}\left(\|E\|_2^2\right),$$

where $s_{n-1} \leq n\sigma_1 \dots \sigma_{n-1}$. Hence we can view s_{n-1} or $n\sigma_1 \dots \sigma_{n-1}$ as first-order condition numbers for absolute perturbations in A.

2.10. The perturbation of a diagonally scaled Jordan block below illustrates that the first-order bound in Remark 2.9 can hold with equality. Let

$$A = \begin{pmatrix} 0 & \alpha_1 & 0 & \dots & 0 \\ \vdots & 0 & \alpha_2 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & & & 0 & \alpha_{n-1} \\ 0 & 0 & \dots & \dots & 0 \end{pmatrix}, \qquad E = \epsilon e_n e_1^*,$$

where $|\epsilon| \leq 1$ and $\alpha_i > 0$, $1 \leq i \leq n-1$. Then $|\det(A+E) - \det(A)| = \alpha_1 \dots \alpha_{n-1}\epsilon$. Since the singular values of A are 0 and $\alpha_i > 0$, $1 \leq i \leq n-1$, we obtain $|\det(A+E) - \det(A)| = s_{n-1}\|E\|_2$.

Replacing the singular values in Corollary 2.7 by powers of $\|A\|_2$ gives the simpler, but weaker bounds below.

COROLLARY 2.11 (normwise bounds). [illegible] A [illegible] E [illegible] $n \times n$ [illegible]

$$\begin{aligned} |\det(A+E) - \det(A)| &\leq \sum_{i=1}^{n} \binom{n}{i} \|A\|_2^{n-i} \|E\|_2^i \\ &= (\|A\|_2 + \|E\|_2)^n - \|A\|_2^n. \end{aligned}$$

[illegible] $\operatorname{rank}(A) = r$ [illegible] $1 \leq r \leq n-1$ [illegible]

$$\begin{aligned} |\det(A+E)| &\leq \|E\|_2^{n-r} \sum_{i=0}^{r} \binom{r}{i} \|A\|_2^{r-i} \|E\|_2^i \\ &= \|E\|_2^{n-r} (\|A\|_2 + \|E\|_2)^r. \end{aligned}$$

[illegible]. This follows from Corollary 2.7 and $s_{n-i} \leq \binom{n}{n-i}\|A\|_2^{n-i} = \binom{n}{i}\|A\|_2^{n-i}$, $1 \leq i \leq n-1$. ∎

A bound similar to the one in Corollary 2.11 was already derived in [1, section 20], [2, Problem I.6.11], [3, Theorem 4.7] for any p-norm, by taking Fréchet derivatives of wedge products. Below we give a basic proof from first principles for the two-norm.

THEOREM 2.12 (section 20 in [1], problem I.6.11 in [2], Theorem 4.7 in [3]). [illegible] A [illegible] E [illegible] $n \times n$ [illegible]

$$|\det(A+E) - \det(A)| \leq n\|E\|_2 \max\{\|A\|_2, \|A+E\|_2\}^{n-1}.$$

[illegible]. We first show the statement for a diagonal matrix. That is, if $D = \operatorname{diag}(\delta_1 \ \dots \ \delta_n)$ is diagonal, then

$$\det(D+F) = \det(D) + z, \qquad \text{where} \quad |z| \leq n\|F\|_2 \max\{\|D\|_2, \|D+F\|_2\}^{n-1}.$$

The proof is by induction. For $n = 2$

$$D = \begin{pmatrix} \delta_1 & \\ & \delta_2 \end{pmatrix}, \qquad F = \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix},$$

and

$$z \equiv \det(D+F) - \det(D) = \delta_1 f_{22} + \det \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & \delta_2 + f_{22} \end{pmatrix}.$$

Lemma 2.5 implies

$$|z| \le \|F\|_2\|D\|_2 + \left\|\begin{pmatrix} f_{11} \\ f_{21}\end{pmatrix}\right\|_2 \left\|\begin{pmatrix} f_{12} \\ \delta_2 + f_{22}\end{pmatrix}\right\|_2 \le \|F\|_2\|D\|_2 + \|F\|_2\|D+F\|_2$$
$$\le 2\|F\|_2 \max\{\|D\|_2, \|D+F\|_2\}.$$

This completes the induction basis. Assuming the statement is true for matrices of order $n-1$, we show that it is also true for matrices of order n. As in the proof of Theorem 2.3, let $D^{(1)} \equiv \operatorname{diag}\begin{pmatrix}0 & \delta_2 & \dots & \delta_n\end{pmatrix}$ be the matrix obtained from D by replacing δ_1 with 0, and apply Lemma 2.1 to conclude

$$\det(D+F) = \delta_1 \det(D_1 + F_1) + \det(D^{(1)} + F).$$

Since $D_1 + F_1$ is a matrix of order $n-1$, the induction hypothesis applies and gives $\det(D_1 + F_1) = \det(D_1) + z_1$, where

$$|z_1| \le (n-1)\|F_1\|_2 \max\{\|D_1\|_2, \|D_1 + F_1\|_2\}^{n-2}$$
$$\le (n-1)\|F\|_2 \max\{\|D\|_2, \|D+F\|_2\}^{n-2}.$$

Substitute the above expression into the expansion for $\det(D+F)$ to obtain

$$z \equiv \det(D+F) - \det(D) = \delta_1 z_1 + \det(D^{(1)} + F),$$

where $|\delta_1 z_1| \le (n-1)\|F\|_2 \max\{\|D\|_2, \|D+F\|_2\}^{n-1}$. Applying Lemma 2.5 to $\det(D^{(1)} + F)$ yields

$$\det(D^{(1)} + F) \le \|Fe_1\|_2 \prod_{i=2}^{n} \|(D+F)e_i\|_2 \le \|F\|_2\|D+F\|_2^{n-1}.$$

Therefore we have proved the theorem for diagonal matrices D.

To prove the theorem for general matrices A, let $A = U\Sigma V^*$ be a SVD of A. Then $\det(A+E) = \det(UV^*)\det(\Sigma + F)$, where $F \equiv U^*EV$. Since Σ is diagonal, $\det(\Sigma + F) = \det(\Sigma) + z$, where

$$|z| \le n\|F\|_2 \max\{\|\Sigma\|_2, \|\Sigma + F\|_2\}^{n-1} = n\|E\|_2 \max\{\|A\|_2, \|A+E\|_2\}^{n-1}.$$

Hence $\det(A+E) - \det(A) = \det(UV^*)z$, and the result follows from $|\det(UV^*)| = 1$. □

2.3. Relative perturbation bounds. We derive expansions for relative perturbations of determinants, as well as relative perturbation bounds that improve existing bounds.

THEOREM 2.13 (expansion). [illegible] A [illegible] E [illegible] $n \times n$ [illegible] A [illegible]

$$\frac{\det(A+E) - \det(A)}{\det(A)} = \det(A^{-1}E) + S_1 + \cdots + S_{n-1},$$

[illegible]

$$S_k \equiv \sum_{1 \le i_1 < \cdots < i_k \le n} \det((A^{-1}E)_{i_1 \dots i_k}), \qquad 1 \le k \le n-1.$$

. Write $\det(A+E) = \det(A)\det(I + A^{-1}E)$ and apply Theorem 2.3 to $\det(I + A^{-1}E)$. □

COROLLARY 2.14 (relative perturbation bound). . A . E $n \times n$ . . . A.

$$\frac{|\det(A+E) - \det(A)|}{|\det(A)|} \leq \left(\kappa \frac{\|E\|_2}{\|A\|_2} + 1\right)^n - 1,$$

. $\kappa \equiv \|A\|_2 \|A^{-1}\|_2$

. Apply Corollary 2.7 to

$$\frac{|\det(A+E) - \det(A)|}{|\det(A)|} = |\det(I + A^{-1}E) - \det(I)|,$$

and bound $\|A^{-1}E\|_2 \leq \kappa \|E\|_2 / \|A\|_2$. □

. 2.15. Corollary 2.14 is more general and tighter than the following bound from [4, (1.6)], [5, Problem 14.15]:

$$\frac{|\det(A+E) - \det(A)|}{|\det(A)|} \leq \frac{n\kappa \|E\|_2/\|A\|_2}{1 - n\kappa \|E\|_2/\|A\|_2},$$

which holds only for $n\kappa\|E\|_2/\|A\|_2 < 1$. This is true because of the following. With $q \equiv \|A^{-1}\|_2 \|E\|_2 = \kappa \|E\|_2/\|A\|_2$ we can write the first term in the bound of Corollary 2.14 as

$$(q+1)^n = \sum_{i=0}^{n} \binom{n}{i} q^i \leq \sum_{i=0}^{n} n^i q^i \leq \sum_{i=0}^{\infty} (nq)^i.$$

If $nq < 1$, then $\sum_{i=0}^{\infty} (nq)^i = \frac{1}{1-nq}$, so that

$$(q+1)^n - 1 \leq \frac{1}{1-nq} - 1 = \frac{nq}{1-nq}.$$

This implies for the bound in Corollary 2.14

$$\left(\kappa \frac{\|E\|_2}{\|A\|_2} + 1\right)^n - 1 \leq \frac{n\kappa ||E\|_2/\|A\|_2}{1 - n\kappa \|E\|_2/\|A\|_2},$$

where the last expression is the bound in [4, inequality (1.6)], [5, Problem 14.15].

2.4. Local sensitivity. We derive a local condition number for determinants from directional derivatives. The directional derivative for $\det(A)$ in the direction E is $\frac{d^k}{dx^k} \det(A + xE)$.

Although we derive the expressions below from the expansion in Theorem 2.3, we could have also used the expression for derivatives of $A(x)$ in [7, equation (6.5.9)].

THEOREM 2.16. . A . E $n \times n$ . . . $F \equiv U^*EV$. x .

$$\det(A + xE) = \sum_{i=1}^{n} S_{n-i} x^i + \det(A),$$

.

$$S_0 \equiv \det(E), \qquad S_k \equiv \det(UV^*) \sum_{1 \leq i_1 < \cdots < i_k \leq n} \sigma_{i_1} \cdots \sigma_{i_k} \det(F_{i_1 \ldots i_k}), \quad 1 \leq k \leq n-1,$$

$$\frac{d^k}{dx^k}\det(A+xE)|_{x=0} = k!S_{n-k}, \qquad 1 \le k \le n.$$

. If $D = \operatorname{diag}(\delta_1 \ \dots \ \delta_n)$ is a diagonal matrix, then Theorem 2.3 implies $\det(D+xF) = \det(xF)+\tilde{S}_1+\cdots+\tilde{S}_{n-1}+\det(D)$, where $\det(xF) = x^n\det(F) = x^nS_0$ and

$$\tilde{S}_k = \sum_{1\le i_1<\cdots<i_k\le n} \delta_{i_1}\cdots\delta_{i_k}\det(xF_{i_1\dots i_k}) = x^{n-k}S_k.$$

To derive the expansion for a general matrix, use the SVD as in Corollary 2.4. □

The first derivative gives the local condition number of the determinant with regard to small perturbations.

COROLLARY 2.17 (local condition number). . A . E . $n\times n$ . . . x . . . .

$$\left|\frac{d}{dx}\det(A+xE)|_{x=0}\right| \le s_{n-1}\|E\|_2, \qquad \text{where} \quad s_{n-1} \le n\sigma_1\dots\sigma_{n-1}.$$

. Theorem 2.16 implies for the first derivative

$$\frac{d}{dx}\det(A+xE)|_{x=0} = \det(UV^*)\sum_{1\le i_1<\cdots<i_{n-1}\le n}\sigma_{i_1}\cdots\sigma_{i_{n-1}}\det(F_{i_1\dots i_{n-1}}),$$

where $F_{i_1\dots i_{n-1}}$ is a diagonal element of F. Lemma 2.5 implies $|\det(F_{i_1\dots i_{n-1}})| \le \|F\|_2 = \|E\|_2$. □

Corollary 2.17 shows that the sensitivity of $\det(A)$ to small perturbations in any direction E is determined by s_{n-1} or $n\sigma_1\dots\sigma_{n-1}$. A comparison with Remark 2.9 shows that the local condition number for $\det(A)$ is identical to the first-order condition number.

3. Characteristic polynomial. Based on the determinant results in section 2, we derive absolute perturbation bounds for the coefficients of the characteristic polynomial for general matrices (section 3.1) and normal matrices (section 3.2), as well as simpler, but weaker normwise bounds (section 3.3).

Applying Theorem 2.3 to the characteristic polynomial

$$\det(\lambda I - A) = \lambda^n + c_1\lambda^{n-1} + \cdots + c_{n-1}\lambda + c_n$$

of the $n\times n$ matrix A gives the well-known expressions [6, Theorem 1.2.12]

$$c_{n-k} = (-1)^{n-k}\sum_{1\le i_1<\dots<i_k\le n}\det(A_{i_1\dots i_k}), \quad 0 \le k \le n-1,$$

where $A_{i_1\dots i_k}$ is the principal submatrix of order $n-k$ obtained by deleting rows and columns $i_1\dots i_k$ of A. The characteristic polynomial of the perturbed matrix $A+E$ is

$$\det(\lambda I - (A+E)) = \lambda^n + \tilde{c}_1\lambda^{n-1} + \cdots + \tilde{c}_{n-1}\lambda + \tilde{c}_n,$$

where $\tilde{c}_n = (-1)^n\det(A+E)$ and

$$\tilde{c}_{n-k} = (-1)^{n-k}\sum_{1\le i_1<\dots<i_k\le n}\det(A_{i_1\dots i_k}+E_{i_1\dots i_k}), \qquad 1 \le k \le n-1.$$

The following example illustrates that products of singular values play an important role in the conditioning of the coefficients c_k.

EXAMPLE 3.1 (companion matrices). The $n \times n$ matrix

$$A = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \dots & \alpha_n \\ \eta & 0 & \dots & \dots & 0 \\ 0 & \eta & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \eta & 0 \end{pmatrix}, \qquad \eta > 0,$$

is a multiple of a companion matrix, and let $E = e_1 \begin{pmatrix} \epsilon & \dots & \epsilon \end{pmatrix}$ with $\epsilon > 0$. The respective coefficients of the characteristic polynomials of A and $A+E$ are [5, section 28.6]

$$c_i = \alpha_i \eta^{i-1}, \qquad \tilde{c}_i = (\alpha_i + \epsilon)\eta^{i-1}, \qquad 1 \le i \le n.$$

Then $|\tilde{c}_i - c_i| = \epsilon\eta^{i-1}$, $1 \le i \le n$. The singular values of A are [5, section 28.6]

$$\sigma_1^2 = \frac{1}{2}\left(\alpha + \sqrt{\alpha^2 - 4|\alpha_n|^2}\right), \qquad \sigma_n^2 = \frac{1}{2}\left(\alpha - \sqrt{\alpha^2 - 4|\alpha_n|^2}\right),$$

where $\alpha \equiv 1 + |\alpha_1|^2 + \dots + |\alpha_n|^2$, and $\sigma_i = \eta$, $2 \le i \le n-1$. Therefore the conditioning of the coefficients c_k is determined by products of singular values.

The products of singular values in our perturbation bounds are expressed in terms of elementary symmetric functions of only the largest singular values of A.

DEFINITION 3.2 (elementary symmetric functions in the largest singular values). Let A be $n \times n$, with singular values $\sigma_1 \ge \dots \ge \sigma_n$. Then

$$s_0^{(k)} \equiv 1, \qquad s_j^{(k)} \equiv \sum_{1 \le i_1 < \dots < i_j \le k} \sigma_{i_1} \dots \sigma_{i_j}, \qquad 1 \le j \le k, \quad 1 \le k \le n,$$

are the elementary symmetric functions in the largest k singular values of A. We set $s_j^{(n)} = s_j$, where s_j is the jth elementary symmetric function in all singular values of A.

3.1. General matrices. We use the determinant expansion in Corollary 2.4 to derive perturbation bounds for coefficients c_k of general matrices.

THEOREM 3.3 (general matrices). Let A and E be $n \times n$, with $\|E\|_2 < \sigma_n$. Then

$$|\tilde{c}_k - c_k| \le \binom{n}{k} \sum_{i=1}^{k} s_{k-i}^{(k)} \|E\|_2^i, \qquad 1 \le k \le n.$$

If $\operatorname{rank}(A) = r$, with $1 \le r \le n-1$, then

$$|\tilde{c}_k - c_k| \le \binom{n}{k} \|E\|_2^{k-r} \sum_{i=0}^{r} s_{r-i}^{(k)} \|E\|_2^i, \qquad r+1 \le k \le n.$$

PROOF. In the perturbed coefficient

$$\tilde{c}_{n-k} = (-1)^{n-k} \sum_{1 \le i_1 < \dots < i_k \le n} \det(A_{i_1 \dots i_k} + E_{i_1 \dots i_k}),$$

the matrices $A_{i_1\dots i_k} + E_{i_1\dots i_k}$ are of order $n-k$. Fix the indices $i_1,\dots,i_k$; set $B \equiv A_{i_1\dots i_k}$ and $F \equiv E_{i_1\dots i_k}$; and let $\mu_1 \geq \dots \geq \mu_{n-k}$ be the singular values of B. Corollary 2.4 implies $\det(B+F) = \det(B) + \det(F) + S_1 + \dots + S_{n-k-1}$, where

$$S_j = \sum_{1\leq i_1<\dots<i_j\leq n-k} \mu_{i_1}\dots\mu_{i_j}\det(F_{i_1\dots i_j}), \qquad 1 \leq j \leq n-k-1.$$

Since B is a submatrix of A, the singular values interlace [6, Theorem 7.3.9], so that $\sigma_j \geq \mu_j$, $1 \leq j \leq n-k$. With Lemma 2.5 we obtain $|S_j| \leq s_j^{(n-k)}\|E\|_2^{n-k-j}$. Hence $|S_1| + \dots + |S_{n-k-1}| \leq \sum_{i=1}^{n-k} s_{n-k-i}^{(n-k)}\|E\|_2^i$. Summing up the terms associated with all $\binom{n}{k}$ submatrices $A_{i_1\dots i_k} + E_{i_1\dots i_k}$ gives the desired bound for $|\tilde{c}_{n-k} - c_{n-k}|$.

Now suppose $\operatorname{rank}(A) = r \leq n-1$. Since r singular values are nonzero, the elementary symmetric functions $s_j^{(k)}$ in the k largest singular values remain unchanged for $k \leq r$.

Since $n-r$ singular values are equal to zero, all products of $r+1$ or more singular values are zero. Hence for $k \geq r+1$ we have $s_j^{(k)} = 0$ whenever $j \geq r+1$, so that

$$\sum_{i=1}^{k} s_{k-i}^{(k)}\|E\|_2^i = \|E\|_2^{k-r}\sum_{i=0}^{r} s_{r-i}^{(k)}\|E\|_2^i.$$

Moreover, for $j \leq r$ the $s_j^{(k)}$ are functions of the r largest singular values only, so that $s_j^{(k)} = s_j^{(r)}$. Therefore $\sum_{i=0}^{k} s_{k-i}^{(k)}\|E\|_2^i = \|E\|_2^{k-r}\sum_{i=0}^{r} s_{r-i}^{(r)}\|E\|_2^i$, giving the desired bound for $|\tilde{c}_k - c_k|$ when $k \geq r+1$. □

For the two extreme coefficients, Theorem 3.3 produces the expected bounds: In the case of $c_n = (-1)^n\det(A)$, the bound coincides with the determinant bound in Corollary 2.7, while for $c_1 = -\operatorname{trace}(A)$ we obtain $|\tilde{c}_1 - c_1| \leq n\|E\|_2$. Theorem 3.3 shows that the conditioning of c_k with regard to absolute perturbations is determined by the binomial term $\binom{n}{k}$ and the elementary symmetric functions in the k largest singular values. The binomial coefficient is largest for c_k with $k \approx n/2$, because $\binom{n}{n-k} = \binom{n}{k}$, and $\binom{n}{k}$ is monotonically increasing for $k < n/2$. In particular, if n is even, then for $k = n/2$ we have $k\binom{n}{k} \geq k\left(\frac{n}{k}\right)^k = n2^{n/2-1}$.

If $\operatorname{rank}(A) = r \leq n-2$, then the bounds for the coefficients $c_{r+1},\dots,c_n$ contain higher powers of $\|E\|_2$. Hence if $\|E\|_2 < 1$, then the coefficients $c_{r+1},\dots,c_n$ of rank-deficient matrices tend to be better conditioned in the absolute sense.

3.4 (first-order absolute condition numbers for general matrices). Theorem 3.3 implies for $\|E\|_2 < 1$ the first-order bound

$$|\tilde{c}_k - c_k| \leq \binom{n}{k} s_{k-1}^{(k)}\|E\|_2 + \mathcal{O}(\|E\|_2^2), \qquad 1 \leq k \leq n,$$

where $s_{k-1}^{(k)} \leq k\sigma_1\dots\sigma_{k-1}$. Hence we can view $\binom{n}{k}s_{k-1}^{(k)}$ or $\binom{n}{k}k\sigma_1\dots\sigma_{k-1}$ as first-order condition numbers for absolute perturbations in the coefficient c_k.

3.2. Normal matrices. We show that for normal matrices, the conditioning of the coefficients improves because the binomial term is smaller, and the elementary symmetric functions depend on all singular values, not just the largest ones. Note that all statements for normal matrices apply in particular to Hermitian matrices.

THEOREM 3.5 (normal matrices). Let A be an $n \times n$ normal matrix, and let

$$|\tilde{c}_k - c_k| \leq \sum_{i=1}^{k} \binom{n-k+i}{i} s_{k-i}\|E\|_2^i, \qquad 1 \leq k \leq n.$$

[illegible] $E = \epsilon I$ [illegible] $\epsilon > 0$ [illegible]. Since A is normal, it has an eigenvalue decomposition $A = V\Lambda V^*$, where $\Lambda = \operatorname{diag}(\lambda_1 \quad \cdots \quad \lambda_n)$ is complex diagonal, $|\lambda_1| \geq \ldots \geq |\lambda_n|$, and V is unitary. Set $D \equiv \lambda I - \Lambda$ and $F \equiv -V^*EV$, so that $\det(\lambda I - (A+E)) = \det(D+F)$. Theorem 2.3 implies $\det(D+F) = \det(D) + \det(F) + S_1 + \cdots + S_{n-1}$. Substituting $\det(D) = \lambda^n + \sum_{k=1}^n c_k \lambda^{n-k}$ and $\det(D+F) = \lambda^n + \sum_{k=1}^n \tilde{c}_k \lambda^{n-k}$ in the above expansion gives

$$\sum_{k=1}^{n} (\tilde{c}_k - c_k)\lambda^{n-k} = \det(F) + S_1 + \cdots + S_{n-1}.$$

Thus $\tilde{c}_k - c_k$ is equal to the coefficient of λ^{n-k} on the right-hand side, i.e., in $\det(F) + S_1 + \cdots + S_{n-1}$. Since

$$S_{n-j} \equiv \sum_{1 \leq i_1 < \cdots < i_{n-j} \leq n} (\lambda - \lambda_{i_1}) \cdots (\lambda - \lambda_{i_{n-j}}) \det(F_{i_1 \ldots i_{n-j}}), \qquad 1 \leq j \leq n-1,$$

has as highest power λ^{n-j}, the term λ^{n-k} can occur only in $S_{n-k}, \ldots, S_{n-1}$. This means $\tilde{c}_k - c_k$ is the sum of the coefficients of λ^{n-k} in $S_{n-1}, \ldots, S_{n-k}$. To bound the coefficient of λ^{n-k} in S_{n-j} in particular, we first bound all coefficients in S_{n-j}.

Observe that S_{n-j} is a sum of $\binom{n}{n-j}$ products $(\lambda - \lambda_{i_1}) \cdots (\lambda - \lambda_{i_{n-j}})$. For fixed $i_1, \ldots, i_{n-j}$ we can write the product as

$$(\lambda - \lambda_{i_1}) \cdots (\lambda - \lambda_{i_{n-j}}) = \lambda^{n-j} + \gamma_1 \lambda^{n-j-1} + \cdots + \gamma_{n-j-1}\lambda + \gamma_{n-j}.$$

The coefficient γ_l is a sum of $\binom{n-j}{l}$ products $\lambda_{j_1} \ldots \lambda_{j_l}$. Hence S_{n-j} contains $\binom{n}{n-j}\binom{n-j}{l}$ such products. Therefore we can bound $|S_{n-j}|$ by a sum of $\binom{n}{n-j}\binom{n-j}{l}$ products $|\lambda_{j_1}| \ldots |\lambda_{j_l}|$. Since A is normal $|\lambda_i| = \sigma_i$, so that these products are also summands of the elementary symmetric function s_l. The sum s_l contains $\binom{n}{l}$ such summands. Therefore the number of occurrences of s_l in the bound for $|S_{n-j}|$ is $\binom{n}{n-j}\binom{n-j}{l} / \binom{n}{l} = \binom{n-l}{j}$.

Now we are ready to return to the coefficient of λ^{n-k} in particular; it is γ_{k-j}. Applying the above counting argument with $l = k - j$ shows that the coefficient of λ^{n-k} in S_{n-j} is bounded by $\binom{n-k+j}{j} s_{k-j} |\det(F_{i_1 \ldots i_{n-j}})|$. Lemma 2.5 implies $|\det(F_{i_1 \ldots i_{n-j}})| \leq \|F\|_2^j = \|E\|_2^j$. Summing up the contributions from all S_{n-j}, $1 \leq j \leq k$, gives the desired result.

If $E = \epsilon I$, then $F = \epsilon I$ and $\det(F_{i_1 \ldots i_k}) = \epsilon^{n-k} = \|E\|_2^{n-k}$. ∎

[illegible] 3.6 (first-order absolute condition numbers for normal matrices). If A is normal and $\|E\|_2 < 1$, then Theorem 3.5 implies the first-order bound

$$|\tilde{c}_k - c_k| \leq (n-k+1)s_{k-1}\|E\|_2 + \mathcal{O}(\|E\|_2^2), \qquad 1 \leq k \leq n,$$

where $s_{k-1} \leq k|\lambda_1 \ldots \lambda_{k-1}|$. Hence we can view $(n-k+1)s_{k-1}$ or $(n-k+1)k|\lambda_1 \ldots \lambda_{k-1}|$ as first-order condition numbers for absolute perturbations in the coefficient c_k.

For Hermitian positive-define matrices, the bound in Theorem 3.5 can be expressed in terms of the coefficients c_k.

COROLLARY 3.7 (Hermitian positive-definite matrices). [illegible] $n \times n$ [illegible] A [illegible]

$$|\tilde{c}_k - c_k| \leq \sum_{i=1}^{k} \binom{n-k+i}{i} |c_{k-i}| \|E\|_2^i, \qquad 1 \leq k \leq n.$$

. The coefficients c_k are also elementary symmetric functions in the eigenvalues [6, section 1.2], and the eigenvalue of a Hermitian positive-definite is equal to the singular values. Thus $c_k = (-1)^k s_k$, and the result follows from Theorem 3.5. ∎

To first order, the conditioning of coefficient c_k is determined by the magnitude of the preceding coefficient, $|c_{k-1}|$. As in Corollary 2.8, the matrix $A+E$ in Corollary 3.7 does not have to be Hermitian positive-definite, because E can be arbitrary. Below we illustrate that one cannot use the expression in Corollary 3.7 for indefinite matrices; that is, positive-definiteness of A is crucial for the expression in Corollary 3.7.

3.8. Corollary 3.7 is not valid for indefinite Hermitian matrices and in particular matrices with zero trace.

To see this, let

$$A = \begin{pmatrix} \alpha & \\ & -\alpha \end{pmatrix}, \qquad \tilde{A} = \begin{pmatrix} \alpha - \epsilon & \\ & -\alpha + \epsilon \end{pmatrix},$$

where $\alpha > 0$ and $\epsilon > 0$. The characteristic polynomials are

$$\det(\lambda I - A) = \lambda^2 - \alpha^2, \qquad (\lambda I - (A + E)) = \lambda^2 - (\alpha - \epsilon)^2,$$

so that $\tilde{c}_2 - c_2 = 2\alpha\epsilon - \epsilon^2$. However, $|\tilde{c}_2 - c_2|$ cannot be bounded in terms of c_1, as required by Corollary 3.7, because $c_1 = 0$.

3.3. Normwise bounds. Replacing the singular value products by powers of $\|A\|_2$ gives the following simpler, but weaker bounds.

COROLLARY 3.9 (normwise bounds). . A . E . $n \times n$.

$$\begin{aligned} |\tilde{c}_k - c_k| &\le k\binom{n}{k} \sum_{i=1}^{k} \binom{k}{i} \|A\|_2^{k-i} \|E\|_2^i, \\ &= \binom{n}{k} \left((\|A\|_2 + \|E\|_2)^k - \|A\|^k \right), \qquad 1 \le k \le n. \end{aligned}$$

. $\operatorname{rank}(A) = r$ . . $1 \le r \le n-1$.

$$\begin{aligned} |\tilde{c}_k - c_k| &\le k\binom{n}{k} \|E\|_2^{k-r} \sum_{i=1}^{r} \binom{k}{i} \|A\|_2^{r-i} \|E\|_2^i, \\ &= \binom{n}{k} \|E\|_2^{k-r} \left((\|A\|_2 + \|E\|_2)^r - \|A\|^r \right), \qquad r+1 \le k \le n. \end{aligned}$$

. This follows from Theorem 3.3 and

$$s_{k-i}^{(k)} \le \binom{k}{k-i} \|A\|_2^{k-i} = \binom{k}{i} \|A\|_2^{k-i}, \qquad 1 \le i \le k-1. \quad ∎$$

A similar bound was was already derived in [1, section 20] and [2, Problem I.6.11] for any p-norm, by taking Fréchet derivatives of wedge products. Below we give a basic proof from first principles for the two-norm.

THEOREM 3.10 (section 20 in [1], problem I.6.11 in [2]). . A . E . $n \times n$.

$$|\tilde{c}_k - c_k| \le k\binom{n}{k} \|E\|_2 \max\{\|A\|_2, \|A + E\|_2\}^{k-1}, \qquad 1 \le k \le n.$$

. As in the proof of Theorem 3.3, we use

$$\tilde{c}_{n-k} = (-1)^{n-k} \sum_{1 \le i_1 < \ldots < i_k \le n} \det(A_{i_1 \ldots i_k} + E_{i_1 \ldots i_k}).$$

This gives for the absolute error

$$|\tilde{c}_{n-k} - c_{n-k}| \le \sum_{1 \le i_1 < \ldots < i_k \le n} |\det(A_{i_1 \ldots i_k} + E_{i_1 \ldots i_k}) - \det(A_{i_1 \ldots i_k})|.$$

Theorem 2.12 implies that $|\det(A_{i_1 \ldots i_k} + E_{i_1 \ldots i_k}) - \det(A_{i_1 \ldots i_k})|$ is bounded by

$$(n-k)\|E_{i_1 \ldots i_k}\|_2 \max\{\|A_{i_1 \ldots i_k}\|_2, \|(A+E)_{i_1 \ldots i_k}\|_2\}^{n-k-1}.$$

Bounding the principal submatrices by the norms of the respective matrices and recognizing that the sum contains $\binom{n}{n-k}$ summands yields the desired bound. ∎

Acknowledgment. We thank Dean Lean for many helpful discussions and for bringing this problem to our attention.

REFERENCES

[1] R. Bhatia, *Perturbation Bounds for Matrix Eigenvalues*, Longman Scientific & Technical, New York, 1987.
[2] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.
[3] S. Friedland, *Variation of tensor powers and permanents*, Linear Multilinear Algebra, 12 (1982), pp. 81–98.
[4] S. K. Godunov, A. G. Antonov, O. P. Kiriljuk, and V. I. Kostin, *Guaranteed accuracy in numerical linear algebra*, Math. Appl. 252, Kluwer Academic Publishers, Dordrecht, 1993 (in English).
[5] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[6] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
[7] R. Horn and C. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, London, 1991.
[8] R. Horn, C. Johnson, P. Lancaster, and M. Tismenetsky, *The Theory of Matrices,* 2 *ed.*, Academic Press, Orlando, 1985.
[9] D. Lee and T. Schaefer, *Neutron matter on the lattice with pionless effective field theory*, Phys. Rev. C, 72 (2005), 024006.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 777–782

LDU FACTORIZATION OF NONSINGULAR TOTALLY NONPOSITIVE MATRICES*

RAFAEL CANTÓ†, PLAMEN KOEV‡, BEATRIZ RICARTE†, AND ANA M. URBANO†

Abstract. An $n \times n$ real matrix A is said to be (totally negative) totally nonpositive if every minor is (negative) nonpositive. In this paper, we study the properties of a totally nonpositive matrix and characterize the case of a nonsingular totally nonpositive matrix A, with $a_{11} < 0$ in terms of its LDU factorization ($L(U)$ is a unit lower- (upper-) triangular matrix, respectively, and D is a diagonal matrix). This characterization allows us to significantly reduce the number of minors to be checked in order to decide the total nonpositivity of a nonsingular matrix with a negative $(1, 1)$ entry.

Key words. nonsingular matrix, totally nonpositive matrix, LDU factorization

AMS subject classifications. 65F40, 15A15, 15A23

DOI. 10.1137/060662897

1. Introduction.

Several types of matrices have an important role in the various branches of mathematics and other sciences. A particular case of these matrices are the totally positive matrices, which have a wide variety of applications in approximation theory, numerical mathematics, statistics, economics, computer aided geometric design, and others fields [6, 12].

We recall that a matrix is called [illegible] if all its minors are nonnegative (positive) and are abbreviated as TP and STP, respectively.

If, instead, all minors of a matrix are nonpositive (negative), the matrix is called [illegible] and are abbreviated as t.n.p. and t.n., respectively.

The TP and STP matrices have been studied by several authors ([1, 2, 3, 4, 5, 6, 7, 8, 11]) who have obtained properties and characterizations in terms of the factorizations obtained by Gaussian or Neville elimination that allow one to significantly reduce the number of minors to be checked in order to decide if a matrix is TP or STP.

For t.n. matrices, a characterization in terms of the parameters obtained from Neville elimination is given in [9]. The spectral properties and [illegible] factorizations are analyzed in [5]. [illegible] factorization ($U(L)$) is a unit upper- (lower-) triangular matrix, respectively, and D is a diagonal matrix).

In this paper we extend the characterization of t.n. matrices given in [5] to nonsingular t.n.p. matrices. In particular, we characterize the t.n.p. matrices in terms of the factors of their [illegible] factorization and provide a criteria to determine if a matrix is t.n.p., which requires that only the sign of minors containing contiguous rows or columns, and include the first row or column, respectively, be checked.

*Received by the editors June 13, 2006; accepted for publication (in revised form) by R. Nabben March 7, 2008; published electronically July 3, 2008.

http://www.siam.org/journals/simax/30-2/66289.html

†Institut de Matemàtica Multidisciplinar, Universitat Politècnica de València (UPV), 46071 València, Spain (rcanto@mat.upv.es, bearibe@mat.upv.es, amurbano@mat.upv.es). The authors' research was supported by Spanish DGI grant MTM2007-64477 and by the UPV under its research program.

‡Department of Mathematics, North Carolina State University, Raleigh, NC 27695 (pskoev@ncsu.edu).

We follow the notation of [1]. Given $k, n \in \mathbb{N}$, $1 \le k \le n$, $\mathcal{Q}_{k,n}$ denotes the set of all increasing sequences of k natural numbers less than or equal to n. If A is an $n \times n$ matrix and $\alpha, \beta \in \mathcal{Q}_{k,n}$, $A[\alpha|\beta]$ denotes the $k \times k$ submatrix of A lying in rows α and columns β. The principal submatrix $A[\alpha|\alpha]$ is abbreviated as $A[\alpha]$.

For a given k, $1 \le k \le n$, a vector $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_k)$ is called [illegible] if $|\epsilon_i| = 1$, $i = 1, 2, \ldots, k$. The matrix A is called sign-regular (strictly sign-regular) with signature ϵ if $\epsilon_k \det A[\alpha|\beta] \ge 0$ (> 0), $\alpha, \beta \in \mathcal{Q}_{k,n}$, $k = 1, 2, \ldots, n$ [1]. In particular, if $\epsilon_i = 1$, $i = 1, 2, \ldots, n$, then A is TP and if $\epsilon_i = -1$, $i = 1, 2, \ldots, n$, then A is t.n.p.

Throughout this paper, an [illegible] will mean the corresponding factorization resulting from Gaussian elimination with no pivoting where L and U are unit lower- and upper-triangular matrices, respectively, and D is diagonal.

2. Characterization of nonsingular t.n.p. matrices by triangular *LDU* factorization. In this section we derive a characterization of t.n.p. matrices in terms of their LDU factorizations. We use the fact that if $A = LDU$ is the LDU factorization of a nonsingular TP matrix, then L (U) is a unit lower- (upper-) triangular TP matrix, respectively, and D is a positive diagonal matrix [1, 2].

THEOREM 2.1. [illegible] A [illegible] $n \times n$ [illegible] $a_{11} < 0$ [illegible] $A = LDU$ [illegible] L (U) [illegible] D [illegible] $(1,1)$ [illegible].

Let P be the permutation matrix $[n, n-1, \ldots, 2, 1]$. From Cauchy–Binet, $G = PAP$ is also a nonsingular t.n.p. matrix.[1]

Let $S = \operatorname{diag}(1, -1, 1, \ldots, \pm 1)$. By [5, Theorem 2.4], $B = SG^{-1}S$ is sign-regular with signature $\epsilon = (1, 1, \ldots, 1, -1)$. Since $a_{11} < 0$,

$$\det B[1, 2, \ldots, n-1] = \det(SG^{-1}S)[1, 2, \ldots, n-1] = \frac{\det G[n]}{\det G} = \frac{a_{11}}{\det A} > 0.$$

Thus we can choose an $x > -1/a_{11}$ so that $C = B + xE_{nn}$ is TP where E_{nn} is the $n \times n$ matrix whose only nonzero entry is 1 in position (n, n). Therefore, $C = L'D'U'$ where L' (U') is a unit lower- (upper-) triangular TP matrix, and D' is a positive diagonal matrix. Consider

$$\begin{aligned} B = C - xE_{nn} &= L'D'U' - xE_{nn} \\ &= \begin{bmatrix} L_1 & 0 \\ l_1 & 1 \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & d_{nn} \end{bmatrix} \begin{bmatrix} U_1 & u_1 \\ 0 & 1 \end{bmatrix} - x \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} L_1 & 0 \\ l_1 & 1 \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & d_{nn} - x \end{bmatrix} \begin{bmatrix} U_1 & u_1 \\ 0 & 1 \end{bmatrix} = L'D''U'. \end{aligned}$$

Since D' is a positive diagonal matrix, D_1 is a positive diagonal matrix and since $\det B < 0$, we have $d_{nn} - x < 0$. Now

$$\begin{aligned} A = PGP &= P(SB^{-1}S)P = P(S(L'D''U')^{-1}S)P \\ &= P(S(U')^{-1}(D'')^{-1}(L')^{-1}S)P \\ &= [P(S(U')^{-1}S)P][P(S(D'')^{-1}S)P][P(S(L')^{-1}S)P] \\ &= (PU''P)(PD'''P)(PL''P) = LDU. \end{aligned}$$

By [5, Theorem 2.4], $L = P(S(U')^{-1}S)P$ is a unit lower-triangular TP matrix, $U = P(S(L')^{-1}S)P$ is a unit upper-triangular TP matrix, and $D = P(S(D'')^{-1}S)P$ is diagonal with all diagonal entries positive except for a negative $(1,1)$ entry. ∎

[1] Note that other permutation similarity transformations do not necessarily preserve the t.n.p. structure.

2.2. Consider the *LDU* factorization of a nonsingular t.n.p. matrix A, with $a_{11} < 0$. The entries of the first column (row) of L (U) are positive because if $l_{j1} = 0$ or $u_{1j} = 0$ for some $j \in \{2, 3, \ldots, n\}$, then $a_{jj} > 0$, which is a contradiction. Moreover, since L (U) is a TP matrix, we have $\det L[j, i|1, j] \geq 0$ ($\det U[1, j|j, i] \geq 0$), and thus $l_{ij} > 0$ ($u_{ji} > 0$) for all $i > j$.

Therefore, L (U) is a unit lower- (upper-) triangular TP matrix with positive entries below (above) the diagonal.

The converse of Theorem 2.1 is not true in general, as the next example shows.

2.3. The matrix

$$A = LDU = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} -3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -3 & -6 & -12 \\ -6 & -10 & -20 \\ -3 & -2 & 1 \end{bmatrix}$$

is not t.n.p. despite the fact that L and U are TP matrices, and D has positive diagonal entries except for the negative $(1, 1)$ entry.

The following theorem gives a necessary condition for a product LDU to be a t.n.p. matrix. Recall that an $n \times n$ triangular matrix is said to be a $\triangle$STP matrix if all of its nontrivial minors are positive.

THEOREM 2.4. [illegible] $A = LDU$ [illegible] $n \times n$ [illegible] $a_{nn} \leq 0$, L (U) [illegible] ($\ldots$) [illegible] ($\ldots$) [illegible] $D = \operatorname{diag}(-d_1, d_2, \ldots, d_n)$ [illegible] $d_i > 0$, $i = 1, 2, \ldots, n$ [illegible] A [illegible].

[illegible]. First consider the case $a_{nn} < 0$. Since L is a unit lower-triangular TP matrix, it can be written as a product of bidiagonal lower-triangular matrices [10] in the form:

$$L = (E_n(m_{n1})E_{n-1}(m_{n-1,1}) \cdots E_2(m_{21}))\,(E_n(m_{n2}) \cdots E_3(m_{32})) \cdots E_n(m_{n,n-1}),$$

where m_{ij} are the multipliers of the Neville elimination of L and $E_i(x)$ is a bidiagonal matrix which differs from the identity only in its $(i, i-1)$ entry x.

Since L is TP, we have $m_{ij} \geq 0$. If we replace the zero multipliers m_{ij} in L by $\delta > 0$, then the resulting matrix, $L(\delta)$ is $\triangle$STP [10]. Analogously, from U we obtain a $\triangle$STP matrix $U(\delta)$.

Now consider $A(\delta) = L(\delta)DU(\delta)$. From Cauchy–Binet,

$$A(\delta) = A + \left[\begin{array}{c|ccc} 0 & 0 & \ldots & 0 \\ \hline 0 & & & \\ \vdots & & p_{ij}(\delta) & \\ 0 & & & \end{array}\right],$$

where $p_{ij}(\delta)$ are polynomials in δ with nonnegative coefficients such that

$$\lim_{\delta \to 0} p_{ij}(\delta) = 0.$$

Since $a_{nn} < 0$ and $A(\delta)(n, n) = a_{nn} + p_{nn}(\delta)$, there exists δ_0 such that $A(\delta)(n, n) < 0$ for all $\delta < \delta_0$. By [5, Theorem 4.2], $A(\delta)$ is a t.n. matrix for all $\delta < \delta_0$ and, in particular,

$$\det A(\delta)[\alpha|\beta] < 0 \qquad \forall\ \alpha, \beta \in \mathcal{Q}_{k,n}, \quad k = 1, 2, \ldots, n.$$

In turn

$$\det A[\alpha|\beta] = \lim_{\delta \to 0} \det A(\delta)[\alpha|\beta] \leq 0 \qquad \forall\ \alpha, \beta \in \mathcal{Q}_{k,n}, \quad k = 1, 2, \ldots, n,$$

that is, A is t.n.p.

Now, suppose that $a_{nn} = 0$. Then the (n, n) entry of $B = A - xE_{nn}$ is $-x$. We have

$$B = A - xE_{nn} = LDU - xE_{nn} = L \begin{bmatrix} -d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n - x \end{bmatrix} U.$$

Therefore for $0 < x < d_n$, by applying the previous case, B is a t.n.p. matrix.

By construction,

$$\det A[\alpha|\beta] = \det B[\alpha|\beta] \leq 0 \quad \forall\, \alpha, \beta \in \mathcal{Q}_{k,n}, \quad k = 1, 2, \ldots, n, \quad n \notin \alpha \cap \beta,$$

and consequently

$$\det A[\{\alpha, n\}|\{\beta, n\}] = \det B[\{\alpha, n\}|\{\beta, n\}] + x \det A[\alpha|\beta] \leq 0.$$

Thus, again, A is t.n.p. □

Combining Theorems 2.1 and 2.4 we obtain the following result.

THEOREM 2.5. [illegible] $n \times n$ [illegible] A [illegible] $a_{11} < 0$ [illegible] $a_{nn} \leq 0$, [illegible] $A = LDU$ [illegible] L (U) [illegible] ([illegible]) [illegible] ([illegible]) [illegible] D [illegible] $(1, 1)$ [illegible]

[illegible] 2.6. By [5, Corollary 4.3], an $n \times n$ matrix A with $a_{nn} < 0$ is t.n. if and only if $A = LDU$ where L (U) is a unit lower- (upper-) $\triangle$STP matrix, and D is a diagonal matrix with all diagonal entries positive except for the negative $(1, 1)$ entry.

3. Some properties of nonsingular t.n.p. matrices. In this section we show some properties of nonsingular t.n.p. matrices analogous to those satisfied by nonsingular TP matrices. From Theorem 2.1 we obtain the following result.

PROPOSITION 3.1. [illegible] A [illegible] $n \times n$ [illegible] [illegible] $a_{11} < 0$, [illegible] $\det A[1, \alpha] < 0$ [illegible] $\alpha \subset \{2, 3, \ldots, n\}$

By Proposition 3.1, the leading principal minors of a nonsingular t.n.p. matrix A with $a_{11} < 0$ are less than zero, that is, $\det A[1, 2, \ldots, k] < 0$ for $k = 1, 2, 3, \ldots, n$. It is not difficult, but is tedious, to prove that this fact is also satisfied when $a_{11} = 0$ and $k = 2, 3, \ldots, n$.

When $a_{11} = 0$ but $a_{nn} < 0$, we have that, PAP, where P is the permutation matrix $[n, n-1, \ldots, 2, 1]$, is a nonsingular t.n.p. matrix with a nonzero $(1, 1)$ entry. So, the matrix PAP admits an LDU factorization, and

$$A = PLDUP = (PLP)(PDP)(PUP) = \hat{U}\hat{D}\hat{L}.$$

Consequently, A admits a $\hat{U}\hat{D}\hat{L}$ factorization where $\hat{L}$ $(\hat{U})$ is a unit lower- (upper-) triangular TP matrix and $\hat{D}$ is a diagonal matrix with positive diagonal entries except for a negative (n, n) entry. A triangular factorization of this type has been obtained in [5, Theorem 4.1] for t.n. matrices, with L (U) being a $\triangle$STP matrix.

Now, from Remark 2.2 we prove the following result about the entries of a nonsingular t.n.p. matrix A with $a_{11} < 0$.

PROPOSITION 3.2. [illegible] A [illegible] $n \times n$ [illegible] [illegible] $a_{11} < 0$, [illegible] $a_{ij} < 0$ [illegible] $i, j = 1, 2, \ldots, n$, [illegible] $(i, j) \neq (n, n)$

[illegible]. By Remark 2.2 the entries of the first row and the first column of A are negative.

Now suppose $a_{ij} = 0$ where $(i,j) \neq (n,n)$ and (say) $i \leq j$ (the case $i \geq j$ being analogous). Then $\det A[1,i|j,j+s] \leq 0$ implies $a_{i,j+s} = 0$ for $s = 1,2,\ldots,n-j$. In turn $\det A[i,t|1,j+s] = 0$ implies that $a_{t,j+s} = 0$ for all $t = i+1, i+2, \ldots, n$, $s = 0,1,2,\ldots,n-j$. In particular, $A[n-1,n|n-1,n] = 0$. From the LDU factorization $A = LDU$ we get

$$A[n-1,n|n-1,n] = L[n-1,n|1,2,\ldots,n] \cdot D \cdot U[1,2,\ldots,n|n-1,n] = 0,$$

which is impossible, since $\operatorname{rank}(L[n-1,n|1,2,\ldots,n]) = \operatorname{rank}(U[1,2,\ldots,n|n-1,n]) = 2$ and $\operatorname{rank}(D) = n$. □

We note that this result still holds when A is a nonsingular t.n.p. matrix with $a_{11} = 0$.

Now, by Propositions 3.1 and 3.2, we prove an important property of nonsingular t.n.p. matrices analogous to the one of nonsingular TP matrices [1, Corollary 3.8].

PROPOSITION 3.3. [illegible] A [illegible] $n \times n$ [illegible] $a_{11} < 0$ [illegible] $a_{nn} < 0$ [illegible] $\det A[\alpha] < 0$ [illegible] $\alpha \in \mathcal{Q}_{k,n}$, $k = 1,2,\ldots,n$ [illegible]. We proceed by induction on the cardinality of α. If $|\alpha| = 1$, there is nothing to prove by Proposition 3.2.

Assume that the result holds for $|\alpha| = k-1$, and consider $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_k)$. By Proposition 3.1, $\det A[1,\alpha] < 0$, and using Sylvester's identity,

$$\det A[1,\alpha] = \frac{\det A[1,\alpha_1,\alpha_2,\ldots,\alpha_{k-1}] \det A[\alpha_1,\alpha_2,\ldots,\alpha_k]}{\det A[\alpha_1,\alpha_2,\ldots,\alpha_{k-1}]} - \frac{\det A[1,\alpha_1,\ldots,\alpha_{k-1}|\alpha_1,\alpha_2,\ldots,\alpha_k] \det A[\alpha_1,\alpha_2,\ldots,\alpha_k|1,\alpha_1,\ldots,\alpha_{k-1}]}{\det A[\alpha_1,\alpha_2,\ldots,\alpha_{k-1}]},$$

we conclude that $\det A[\alpha] < 0$, which concludes the induction. □

4. A simplified characterization of nonsingular t.n.p. matrices by minors. The following result provides a simple characterization of nonsingular t.n.p. matrices in terms of the sign of some of their minors when the $(1,1)$ entry is negative.

THEOREM 4.1. [illegible] A [illegible] $n \times n$ [illegible] (n,n) [illegible] A [illegible] $k = 1,2,\ldots,n$ [illegible]

$$\det A[\alpha|1,2,\ldots,k] \leq 0 \quad \text{[illegible] } \alpha \in \mathcal{Q}_{k,n}, \tag{4.1}$$

$$\det A[1,2,\ldots,k|\beta] \leq 0 \quad \text{[illegible] } \beta \in \mathcal{Q}_{k,n}, \tag{4.2}$$

$$\det A[1,2,\ldots,k] < 0. \tag{4.3}$$

[illegible]. If A is t.n.p., then the inequalities (4.1) and (4.2) follow from the definition of a t.n.p. matrix, and (4.3) is given in Proposition 3.1.

Conversely, from (4.3), let $A = LDU$ be its LDU factorization. Then $D = \operatorname{diag}(-d_1, d_2, \ldots, d_n)$ where $d_i > 0$, $i = 1,2,\ldots,n$. Hence, from Cauchy–Binet and (4.1),

$$\begin{aligned}\det(LD)[\alpha|1,2,\ldots,k] &= \det L[\alpha|1,2,\ldots,k](-d_1)d_2 \cdots d_k \\ &= \det A[\alpha|1,2,\ldots,k] \leq 0 \quad \text{for all } \alpha \in \mathcal{Q}_{k,n},\ k = 1,2,\ldots,n.\end{aligned}$$

Therefore, $\det L[\alpha|1,2,\ldots,k] \geq 0$ and, by [1, Corollary 2.2], L is a TP matrix. Moreover, since the entries of the first column of A are negative, the entries of the first column of L are positive. By Remark 2.2, L is a unit lower-triangular TP matrix with

positive entries below the diagonal. Analogously, U is a unit upper-triangular TP matrix with positive entries above the diagonal. Now, by Theorem 2.4, the nonsingular matrix A is t.n.p. □

4.2. By Remark 2.6 and [1, Corollary 2.6], we obtain the following characterization for t.n. matrices in terms of their minors. An $n \times n$ matrix A is t.n. if and only if for each $k = 1, 2, \ldots, n$, the following inequalities hold:

$$\begin{aligned} &\det A[\alpha|1,2,\ldots,k] < 0 \quad \text{for all } \alpha \in \mathcal{Q}_{k,n} \quad \text{with } \ d(\alpha) = 0, \\ &\det A[1,2,\ldots,k|\beta] < 0 \quad \text{for all } \beta \in \mathcal{Q}_{k,n} \quad \text{with } \ d(\beta) = 0. \end{aligned}$$

The characterizations given in Theorem 4.1 and Remark 4.2 for t.n.p. and t.n. matrices are analogous to the results for nonsingular TP and STP matrices given in [8].

REFERENCES

[1] T. Ando, *Totally positive matrices*, Linear Algebra Appl., 90 (1987), pp. 165–219.

[2] C. W. Cryer, *The LU-factorization of totally positive matrices*, Linear Algebra Appl., 7 (1973), pp. 83–92.

[3] S. M. Fallat, A. Herman, M. I. Gekhtman, and C. R. Johnson, *Compressions of totally positive matrices*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 68–80.

[4] S. M. Fallat, C. R. Johnson, and R. L. Smith, *The general totally positive matrix completion problem with few unspecified entries*, Electronic J. Linear Algebra, 7 (2000), pp. 1–20.

[5] S. M. Fallat and P. Van Den Driessche, *On matrices with all minors negative*, Electronic J. Linear Algebra, 7 (2000), pp. 92–99.

[6] M. Gasca and C. A. Micchelli, eds., *Total Positivity and Its Applications*, Math. Appl. 359, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[7] M. Gasca and J. M. Peña, *Total positivity and Neville elimination*, Linear Algebra Appl., 44 (1992), pp. 25–44.

[8] M. Gasca and J. M. Peña, *Total positivity, QR factorization and Neville elimination*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1132–1140.

[9] M. Gasca and J. M. Peña, *A test for strict sign-regularity*, Linear Algebra Appl., 197/198 (1994), pp. 133–142.

[10] M. Gasca and J. M. Peña, *On factorizations of totally positive matrices*, in Total Positivity and Its Applications, M. Gasca and C. A. Micchelli, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996, pp. 109–130.

[11] M. Gassó and J. R. Torregrosa, *A totally positive factorization of rectangular matrices by the Neville elimination*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 986–994.

[12] S. Karlin, *Total Positivity, Vol.* I, Stanford University Press, Stanford, CA, 1968.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 783–804

OPTIMIZATION OF GENERALIZED MEAN-SQUARE ERROR IN NOISY LINEAR ESTIMATION*

WILLIAM W. HAGER† AND JIANGTAO LUO†

Abstract. A class of least squares problems that arises in linear Bayesian estimation is analyzed. The data vector $\mathbf{y}$ is given by the model $\mathbf{y} = \mathbf{P}(\mathbf{H}\boldsymbol{\theta} + \boldsymbol{\eta}) + \mathbf{w}$, where $\mathbf{H}$ is a known matrix, while $\boldsymbol{\theta}$, $\boldsymbol{\eta}$, and $\mathbf{w}$ are uncorrelated random vectors. The goal is to obtain the best estimate for $\boldsymbol{\theta}$ from the measured data. Applications of this estimation problem arise in multisensor data fusion problems and in wireless communication. The unknown matrix $\mathbf{P}$ is chosen to minimize the expected mean-squared error $\mathbf{E}(\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|^2)$ subject to a power constraint "trace $(\mathbf{P}\mathbf{P}^*) \le P$," where $\hat{\boldsymbol{\theta}}$ is the best affine estimate of $\boldsymbol{\theta}$. Earlier work characterized an optimal $\mathbf{P}$ in the case where the noise term $\boldsymbol{\eta}$ vanished, while this paper analyzes the effect of $\boldsymbol{\eta}$, assuming its covariance is a multiple of $\mathbf{I}$. The singular value decomposition of an optimal $\mathbf{P}$ is expressed in the form $\mathbf{V}\boldsymbol{\Sigma}\boldsymbol{\Pi}\mathbf{U}^*$ where $\mathbf{V}$ and $\mathbf{U}$ are unitary matrices related to the covariance of either $\boldsymbol{\theta}$ or $\mathbf{w}$, and singular vectors of $\mathbf{H}$, $\boldsymbol{\Sigma}$ is diagonal, and $\boldsymbol{\Pi}$ is a permutation matrix. The analysis is carried out in two special cases: (i) $\mathbf{H} = \mathbf{I}$ and (ii) covariance of $\boldsymbol{\theta}$ is $\mathbf{I}$. In case (i), $\boldsymbol{\Pi}$ does not depend on the power P. In case (ii), $\boldsymbol{\Pi}$ generally depends on P. The optimal $\boldsymbol{\Pi}$ is determined in the limit as the power tends to zero or infinity; a good approximation to an optimal $\boldsymbol{\Pi}$ is found for general P.

Key words. linear Bayesian estimation, mean-square error, MSE, CDMA systems, wireless communication

AMS subject classifications. 60G35, 93E10, 94A15

DOI. 10.1137/060676830

1. Introduction.

Suppose that $\mathbf{y} \in \mathbb{C}^m$ is a random vector that obeys the model

$$\mathbf{y} = \mathbf{P}(\mathbf{H}\boldsymbol{\theta} + \boldsymbol{\eta}) + \mathbf{w}, \tag{1.1}$$

where $\mathbf{H} \in \mathbb{C}^{n\times l}$ is a known matrix, while $\boldsymbol{\theta} \in \mathbb{C}^l$, $\boldsymbol{\eta} \in \mathbb{C}^n$, and $\mathbf{w} \in \mathbb{C}^m$ are uncorrelated random vectors with the property that $\boldsymbol{\eta}$ and $\mathbf{w}$ have zero mean. The matrix $\mathbf{P} \in \mathbb{C}^{m\times n}$ is a "filter" which is applied to the noisy measurement $\mathbf{H}\boldsymbol{\theta} + \boldsymbol{\eta}$, and which is chosen to achieve an optimal estimate of the signal $\boldsymbol{\theta}$. We consider affine estimators of the form

$$\hat{\boldsymbol{\theta}} = \mathbf{A}\mathbf{y} + \mathbf{a},$$

where $\mathbf{A} \in \mathbb{C}^{l\times m}$ is a constant matrix and $\mathbf{a} \in \mathbb{C}^l$ is a constant vector. For any random vector $\mathbf{v}$, let $\mathbf{C}_v$ denote the covariance defined by

$$\mathbf{C}_v = \mathbf{E}\big((\mathbf{v} - \mathbf{E}(\mathbf{v}))(\mathbf{v} - \mathbf{E}(\mathbf{v}))^*\big),$$

where $\mathbf{E}$ is expectation and * is conjugate transpose. According to [4, Thm. 12.1], the affine estimator that minimizes the expected mean-square error $\mathbf{E}(\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|^2)$ is given by

$$\hat{\boldsymbol{\theta}} = \mathbf{E}(\boldsymbol{\theta}) + (\mathbf{C}_\theta^{-1} + \mathbf{H}^*\mathbf{P}^*\mathbf{C}_{\eta w}^{-1}\mathbf{P}\mathbf{H})^{-1}(\mathbf{P}\mathbf{H})^*\mathbf{C}_{\eta w}^{-1}(\mathbf{y} - \mathbf{P}\mathbf{H}\mathbf{E}(\boldsymbol{\theta})),$$

*Received by the editors December 6, 2006; accepted for publication (in revised form) by A. S. Lewis April 28, 2008; published electronically September 4, 2008. This material is based upon work supported by the National Science Foundation under grants 0203270, 0619080, 0620286, and 0724750.
http://www.siam.org/journals/simax/30-2/67683.html

†Department of Mathematics, University of Florida, Gainesville, FL 32611-8105 (hager@math.ufl.edu, http://www.math.ufl.edu/∼hager, luo@math.ufl.edu, http://www.math.ufl.edu/∼luo3).

where $\mathbf{C}_{\eta w} = \mathbf{P}\mathbf{C}_\eta\mathbf{P}^* + \mathbf{C}_w$. Moreover, the error $\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$ has zero mean and covariance

$$\mathbf{C} = \mathbf{E}\big((\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^*\big) = \big((\mathbf{P}\mathbf{H})^*(\mathbf{C}_w + \mathbf{P}\mathbf{C}_\eta\mathbf{P}^*)^{-1}\mathbf{P}\mathbf{H} + \mathbf{C}_\theta^{-1}\big)^{-1}. \tag{1.2}$$

Since $\hat{\boldsymbol{\theta}}$ depends on $\mathbf{P}$, the estimation error $\mathbf{E}(\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|^2)$ depends on $\mathbf{P}$. The filter $\mathbf{P}$ is chosen to minimize the estimation error, subject to the constraint tr $(\mathbf{P}\mathbf{P}^*) \le P$, where P is a positive scalar and "tr" denotes the trace of a matrix. The constraint tr $(\mathbf{P}\mathbf{P}^*) \le P$ represents a bound on the power associated with $\mathbf{P}$.

Multisensor data fusion problems (see [6, 10, 15] and the references therein) fit within the framework of the model (1.1). In these applications, $\boldsymbol{\theta}$ is a random parameter vector that is being measured by a collection of sensors. The sensor measurements correspond to the observation matrix $\mathbf{H}$. The term $\boldsymbol{\eta}$ in the model (1.1) could represent sensor noise. The output of the observer is sent to the "fusion center" which leads to the final output $\mathbf{y}$ in (1.1). If the dimension of the column space of $\mathbf{P}$ is less than the dimension of the row space, then there is a reduction in dimensionality of the data. The term $\mathbf{w}$ might represent either noise or quantization error in the transmission to the fusion center. The constraint tr $(\mathbf{P}\mathbf{P}^*) \le P$ might also be viewed as a constraint on the amplifier gain to prevent the amplified observations from exceeding the dynamic range of the quantizer.

Another application which fits the model (1.1) concerns spreading sequence optimization for code division multiple access (CDMA) communication systems [12, 13]. In CDMA systems, many users simultaneously share a communication channel. In modeling the uplink (communication from the mobile units to the base station), $\mathbf{y}$ is the signal received at the base station, the jth column of $\mathbf{P}$ is the "spreading sequence" assigned to the jth user, and θ_j is the symbol transmitted from the jth user. $\mathbf{H}$ is a diagonal matrix corresponding to channel gains.

The problem of estimating the channel matrix for a multiple input, single output (MISO) system can be expressed in the form (1.1) as observed in [2]. In this context, there are multiple transmit antennas and a single receiver. The jth column of $\mathbf{P}$ is the training signal to transmit from the jth antenna to obtain the best estimate for the communication channel gains; the matrix $\mathbf{H}$ is the square root of the correlation between the transmit antennas. The noise in the channel gains and in the transmitted signal associated with atmospheric conditions is modeled by $\boldsymbol{\eta}$ and $\mathbf{w}$.

The model (1.1) is related to the channel estimation problem for multiple input, multiple output (MIMO) systems [5, 14]. That is, in [5] it is shown that when $\mathbf{H}$ and $\mathbf{C}_w$ have a special Kronecker product form and when $\mathbf{C}_\eta = \mathbf{0}$, then the covariance of the best channel estimate is a multiple of $\mathbf{C}$ in (1.2). The model (1.1) is loosely connected with joint linear transmitter-receiver design in MIMO communication [8, 9]. In MIMO communication, the precoder $\mathbf{P}$ precedes the channel matrix $\mathbf{H}$. Hence, in the special case $\mathbf{H} = \mathbf{I}$, the model (1.1) corresponds to a MIMO communication channel with two noise terms.

Since $\mathbf{E}(\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|^2)$ is the trace of $\mathbf{C}$, minimizing the trace of the covariance $\mathbf{C}$ is equivalent to minimizing $\mathbf{E}(\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|^2)$. Hence, the $\mathbf{P}$ that minimizes the expected mean-square error $\mathbf{E}(\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|^2)$ is a solution of the problem

$$\begin{aligned} &\min_{\mathbf{P}} \ \text{tr}\ \big((\mathbf{P}\mathbf{H})^*(\mathbf{W} + \mathbf{P}\mathbf{N}\mathbf{P}^*)^{-1}\mathbf{P}\mathbf{H} + \mathbf{T}\big)^{-1} \\ &\text{subject to} \ \ \text{tr}\ (\mathbf{P}\mathbf{P}^*) \le P, \quad \mathbf{P} \in \mathbb{C}^{m\times n}, \end{aligned} \tag{1.3}$$

where $\mathbf{W} = \mathbf{C}_w$, $\mathbf{N} = \mathbf{C}_\eta$, and $\mathbf{T} = \mathbf{C}_\theta^{-1}$. Both $\mathbf{C}_w$ and $\mathbf{C}_\theta$ are assumed positive definite. This holds, for example, if the probability density function associated with

$\boldsymbol{\theta}$ and $\mathbf{w}$ is continuous. The wireless communication applications in [2] correspond to $\mathbf{N} = \mathbf{0}$ and $\mathbf{T} = \mathbf{I}$. The application in [5] corresponds to $\mathbf{N} = \mathbf{0}$ and $\mathbf{H} = \mathbf{I}$. In this paper, we again consider the cases (i) $\mathbf{H} = \mathbf{I}$ or (ii) $\mathbf{T} = \mathbf{I}$; however, the noise covariance $\mathbf{N}$ is no longer zero, but a multiple of $\mathbf{I}$. By a rescaling of the variables $\mathbf{P}$, $\mathbf{H}$, and the power P, there is no loss of generality in assuming that $\mathbf{N} = \mathbf{I}$. In the multisensor data fusion problem studied in [6], it is pointed out that when the sensor noise is uncorrelated with the signal $\boldsymbol{\theta}$ and when the sensor noise is spatially uncorrelated with zero mean, then by appropriate pre- and post-whitening if necessary, there is no loss of generality in assuming that $\mathbf{N} = \mathbf{I}$ and $\mathbf{T} = \mathbf{I}$. Hence, we focus on the problem

$$\begin{aligned} &\min_{\mathbf{P}} \ \operatorname{tr}\left((\mathbf{PH})^*(\mathbf{W}+\mathbf{PP}^*)^{-1}\mathbf{PH}+\mathbf{T}\right)^{-1} \\ &\text{subject to } \ \operatorname{tr}(\mathbf{PP}^*) \le P, \quad \mathbf{P} \in \mathbb{C}^{m\times n}. \end{aligned} \tag{1.4}$$

A unified analysis is developed for (1.4) which handles the special cases (i) or (ii) and which exposes the similarities and differences in these two problems. In both cases, the singular value decomposition of an optimal solution is expressed in the form $\mathbf{V\Sigma\Pi U}^*$ where $\mathbf{V}$ and $\mathbf{U}$ are unitary matrices related to eigenvectors of $\mathbf{W}$ or $\mathbf{T}$ or singular vectors of $\mathbf{H}$, with a specific ordering for the columns described below. The matrix $\mathbf{\Sigma}$ is diagonal, and $\mathbf{\Pi}$ is a permutation matrix. A fundamental difference in these problems is that in case (i) ($\mathbf{H} = \mathbf{I}$), the permutation is independent of P, which is the same result obtained in [5] for $\boldsymbol{\eta} = \mathbf{0}$. For case (ii) ($\mathbf{T} = \mathbf{I}$), $\mathbf{\Pi}$ depends on the choice of P and the singular values of $\mathbf{H}$. When the noise term $\boldsymbol{\eta}$ vanishes and P is large, the permutation arranges the singular values of $\mathbf{H}$ in increasing order, as obtained in [2]; when noise $\boldsymbol{\eta}$ with covariance $\mathbf{I}$ is included in the model, the singular values of $\mathbf{H}$ smaller than one are arranged in decreasing order, while the singular values greater than one are arranged in increasing order (see Theorem 5.1). As a result, depending on the size of the power P and the distribution of the singular values of $\mathbf{H}$, the noise term $\boldsymbol{\eta}$ can have a significant effect on the structure of the optimal solution.

The paper is organized as follows: In section 2 we derive the singular value decomposition $\mathbf{V\Sigma\Pi U}^*$ of a solution to (1.4). Section 3 gives the optimal solution, assuming the permutation $\mathbf{\Pi}$ is known. In section 4, we evaluate $\mathbf{\Pi}$ in the special case $\mathbf{H} = \mathbf{I}$. When $\mathbf{H} \neq \mathbf{I}$, $\mathbf{\Pi}$ depends on P. Section 5 evaluates the permutation in the limit as P tends to infinity, while section 6 analyzes the limit as P tends to zero. Finally, section 7 explores the dependence of the permutation on P using randomly generated test problems. For general P, we present a family of permutations which often contains the optimal $\mathbf{\Pi}$.

Throughout the paper, we use the following notation: $\mathbf{U\Lambda V}^*$ is the singular value decomposition of $\mathbf{H}$ (see Figure 1.1) and $\boldsymbol{\lambda}$ is the diagonal of $\mathbf{\Lambda}$. The diagonal of a rectangular matrix $\mathbf{\Lambda}$ are the entries $\mathbf{\Lambda}_{ii}$, $i = 1, 2, \ldots, \min\{m, n\}$. $\mathbf{V}_w\mathbf{\Omega V}_w^*$ and $\mathbf{V}_t\mathbf{\Theta V}_t^*$ are diagonalizations of the Hermitian matrices $\mathbf{W}$ and $\mathbf{T}$, respectively, while $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ are the diagonals of $\mathbf{\Omega}$ and $\mathbf{\Theta}$. The diagonal elements are ordered as follows:

$$\lambda_i \ge \lambda_{i+1}, \quad \theta_i \le \theta_{i+1}, \quad \text{and} \quad \omega_i \le \omega_{i+1}. \tag{1.5}$$

M denotes the minimum of m and the rank of $\mathbf{H}$. The trace of a matrix is denoted "tr," "*" denotes conjugate transpose, $\mathcal{S}^c$ denotes complement of the set $\mathcal{S}$, and $|\mathcal{S}|$ is the number of elements in $\mathcal{S}$. A diagonal matrix $\mathbf{D}$ is said to be nondegenerate if

Decomposition	Dimension	Description
$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^*$	$n \times l$	Singular value decomposition of observer
$\mathbf{W} = \mathbf{V}_w\mathbf{\Omega}\mathbf{V}_w^*$	$m \times m$	Diagonalization of covariance of $\mathbf{w}$
$\mathbf{T} = \mathbf{V}_t\mathbf{\Theta}\mathbf{V}_t^*$	$l \times l$	Diagonalization of inverse of covariance of $\boldsymbol{\theta}$
$\mathbf{P} = \mathbf{V}_w\mathbf{S}\mathbf{V}_t^*$	$m \times n$	Change of variables when $\mathbf{H} = \mathbf{I}$
$\mathbf{P} = \mathbf{V}_w\mathbf{S}\mathbf{U}^*$	$m \times n$	Change of variables when $\mathbf{\Theta} = \mathbf{I}$

FIG. 1.1. *Summary of decompositions, $l = n$ without loss of generality.*

the following condition is satisfied:

$$d_{ii} \neq d_{jj} > 0 \text{ for all } i \neq j. \tag{1.6}$$

For any matrix $\mathbf{A}$, $\mathrm{Col}_k(\mathbf{A})$ denotes the submatrix formed by the first k columns, while $\mathrm{Prin}_k(\mathbf{A})$ denotes the k by k leading principal submatrix. $\mathcal{P}_m$ is the set of bijections of $\{1, 2, \ldots, m\}$ onto itself (the set of all permutations of the integers between 1 and m).

2. Solution structure. We begin by analyzing the structure of an optimal solution to (1.4). Let us make the following change of variables:

$$\mathbf{P} = \mathbf{V}_w\mathbf{S}\mathbf{U}^* \ \ (\text{if } \mathbf{\Theta} = \mathbf{I}) \quad \text{or} \quad \mathbf{P} = \mathbf{V}_w\mathbf{S}\mathbf{V}_t^* \ \ (\text{if } \mathbf{H} = \mathbf{I}). \tag{2.1}$$

With these substitutions, (1.4) reduces to the following problem in the cases $\mathbf{H} = \mathbf{I}$ or $\mathbf{T} = \mathbf{I}$:

$$\begin{aligned} \min_{\mathbf{S}} \quad & \mathrm{tr}\ ((\mathbf{S}\mathbf{\Lambda})^*(\mathbf{\Omega} + \mathbf{S}\mathbf{S}^*)^{-1}\mathbf{S}\mathbf{\Lambda} + \mathbf{\Theta})^{-1} \\ \text{subject to} \quad & \mathrm{tr}\ (\mathbf{S}\mathbf{S}^*) \leq P, \quad \mathbf{S} \in \mathbb{C}^{m \times n}. \end{aligned} \tag{2.2}$$

If $\mathbf{H} = \mathbf{I}$, then $l = n$. We now show that in general (2.2) can always be transformed to an equivalent problem with $l = n$. Note though that the transformed problem may have zero singular values in $\mathbf{H}$ even when the singular values of the original $\mathbf{H}$ are strictly positive. If $l > n$, then define $\overline{\mathbf{\Lambda}} = \mathrm{Col}_n(\mathbf{\Lambda})$, the submatrix formed by the first n columns of $\mathbf{\Lambda}$, and define

$$\overline{\mathbf{C}} = ((\mathbf{S}\overline{\mathbf{\Lambda}})^*(\mathbf{\Omega} + \mathbf{S}\mathbf{S}^*)^{-1}\mathbf{S}\overline{\mathbf{\Lambda}} + \mathbf{\Theta}_1)^{-1},$$

where $\mathbf{\Theta}_1 = \mathrm{Prin}_n(\mathbf{\Theta})$, the leading n by n principal submatrix of $\mathbf{\Theta}$. Since the last $l - n$ columns of $\mathbf{\Lambda}$ are zero, the covariance matrix

$$\mathbf{C} = ((\mathbf{S}\mathbf{\Lambda})^*(\mathbf{\Omega} + \mathbf{S}\mathbf{S}^*)^{-1}\mathbf{S}\mathbf{\Lambda} + \mathbf{\Theta})^{-1}$$

has the structure

$$\mathbf{C} = \begin{bmatrix} \overline{\mathbf{C}} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Theta}_2 \end{bmatrix},$$

where $\mathbf{\Theta}_2$ is the trailing $l - n$ by $l - n$ submatrix of $\mathbf{\Theta}$. Hence,

$$\mathrm{tr}\ (\mathbf{C}) = \mathrm{tr}\ (\overline{\mathbf{C}}) + \mathrm{tr}\ (\mathbf{\Theta}_2^{-1}),$$

and minimizing the trace of $\mathbf{C}$ is equivalent to minimizing the trace of $\overline{\mathbf{C}}$ (since $\mathbf{\Theta}_2$ does not depend on $\mathbf{S}$).

On the other hand, suppose that $l < n$. Let $\mathbf{\Lambda}_0$ be the matrix obtained by appending $n-l$ columns of zeros to the right side of $\mathbf{\Lambda}$, let $\overline{\mathbf{\Theta}}$ be the matrix obtained by appending $n-l$ trailing ones on the diagonal of $\mathbf{\Theta}$, and define

$$\mathbf{C}_0 = ((\mathbf{S}\mathbf{\Lambda}_0)^*(\mathbf{\Omega} + \mathbf{S}\mathbf{S}^*)^{-1}\mathbf{S}\mathbf{\Lambda}_0 + \overline{\mathbf{\Theta}})^{-1}.$$

The matrix $\mathbf{C}_0$ has the following structure:

$$\mathbf{C}_0 = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

Hence, tr $(\mathbf{C}_0)$ = tr $(\mathbf{C}) + n - l$, and minimizing the trace of $\mathbf{C}$ is equivalent to minimizing the trace of $\mathbf{C}_0$. In either case $l > n$ or $l < n$, we are able to formulate a problem with the associated $\mathbf{\Lambda}$ square and with the same solution as the original problem. Consequently, it is assumed henceforth that $l = n$. We begin by formulating the first-order optimality conditions for (2.2).

LEMMA 2.1. [illegible] (2.2) [illegible] $\boldsymbol{\lambda} > \mathbf{0}$ [illegible] $\mu > 0$ [illegible]

(2.3) $$(\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})\mathbf{\Lambda}\mathbf{M}^{-2}\mathbf{\Lambda}\mathbf{S}^*\mathbf{L}^{-1} = \mu\mathbf{S}^*,$$

[illegible]

(2.4) $$\mathbf{L} = \mathbf{\Omega} + \mathbf{S}\mathbf{S}^* \quad \text{and} \quad \mathbf{M} = \mathbf{\Lambda}\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}\mathbf{\Lambda} + \mathbf{\Theta}.$$

[illegible] $\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$ [illegible] $\mathbf{\Lambda}\mathbf{M}^{-2}\mathbf{\Lambda}$ [illegible]

[illegible]. If $\mathbf{S} = \mathbf{0}$, then the results hold trivially. Suppose that $\mathbf{S} \neq \mathbf{0}$. The first-order necessary optimality conditions are satisfied at any nonzero solution of (2.2) since the gradient of the constraint does not vanish. Hence, if $\mathbf{S}$ is a solution of (2.2), then there exists a $\mu \geq 0$ such that the Fréchet derivative of the Lagrangian vanishes at $\mathbf{S}$. The Lagrangian associated with the optimization problem (2.2) is

(2.5) $$\text{tr } \left((\mathbf{\Lambda}\mathbf{S}^*(\mathbf{\Omega} + \mathbf{S}\mathbf{S}^*)^{-1}\mathbf{S}\mathbf{\Lambda} + \mathbf{\Theta})^{-1} + \mu\mathbf{S}\mathbf{S}^*\right),$$

where the multiplier $\mu \geq 0$ is a real scalar. As shown in the Appendix, when we equate to zero the derivative of the Lagrangian, we obtain (2.3).

We now show that the multiplier μ is strictly positive. Suppose $\mu = 0$. Since $\mathbf{\Omega}$ and $\mathbf{\Theta}$ are positive definite (see section 1), the factors $\mathbf{L}$ and $\mathbf{M}$ in (2.3) are positive definite. Since $\boldsymbol{\lambda} > \mathbf{0}$, $\mathbf{\Lambda}$ is positive definite. If we can show that $(\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})$ is invertible, then $\mu = 0$ implies that $\mathbf{S} = \mathbf{0}$, which is a contradiction. Thus $\mu > 0$.

To show that $(\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})$ is invertible, we apply the matrix modification formula [3]

(2.6) $$(\mathbf{I} + \mathbf{Z}\mathbf{Z}^*)^{-1} = \mathbf{I} - \mathbf{Z}(\mathbf{I} + \mathbf{Z}^*\mathbf{Z})^{-1}\mathbf{Z}^*$$

with $\mathbf{Z}^* = \mathbf{\Omega}^{-1/2}\mathbf{S}$ to obtain

$$\begin{aligned} \mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S} &= \mathbf{I} - \mathbf{S}^*(\mathbf{\Omega} + \mathbf{S}\mathbf{S}^*)^{-1}\mathbf{S} \\ &= \mathbf{I} - \mathbf{S}^*\mathbf{\Omega}^{-1/2}(\mathbf{I} + \mathbf{\Omega}^{-1/2}\mathbf{S}\mathbf{S}^*\mathbf{\Omega}^{-1/2})^{-1}\mathbf{\Omega}^{-1/2}\mathbf{S} \\ &= \mathbf{I} - \mathbf{Z}(\mathbf{I} + \mathbf{Z}^*\mathbf{Z})^{-1}\mathbf{Z}^* = (\mathbf{I} + \mathbf{Z}\mathbf{Z}^*)^{-1} \\ &= (\mathbf{I} + \mathbf{S}^*\mathbf{\Omega}^{-1}\mathbf{S})^{-1}. \end{aligned}$$

Hence, the matrix $\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$ is positive definite and invertible. This completes the proof that $\mu > 0$.

We multiply (2.3) by $\mathbf{S}$ to obtain

$$\mu\mathbf{S}^*\mathbf{S} = (\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})\mathbf{\Lambda}\mathbf{M}^{-2}\mathbf{\Lambda}\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}. \tag{2.7}$$

Forming the conjugate transpose of (2.7) gives

$$\mu\mathbf{S}^*\mathbf{S} = \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}\mathbf{\Lambda}\mathbf{M}^{-2}\mathbf{\Lambda}(\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}). \tag{2.8}$$

Equating the right sides of (2.7) and (2.8) yields

$$(\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})(\mathbf{\Lambda}\mathbf{M}^{-2}\mathbf{\Lambda}) = (\mathbf{\Lambda}\mathbf{M}^{-2}\mathbf{\Lambda})(\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}). \tag{2.9}$$

Hence, the matrices $\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$ and $\mathbf{\Lambda}\mathbf{M}^{-2}\mathbf{\Lambda}$ commute. □

We now present cases where (2.2) has a solution with at most one nonzero in each row and column.

LEMMA 2.2. [illegible] $\mathbf{S}$ [illegible] (2.2) [illegible] $\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$ [illegible] $\mathrm{Prin}_k(\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})$ [illegible] $k > 0$ [illegible] $\mathbf{\Omega}$ [illegible] $\mathbf{\Lambda}$ [illegible] $\mathrm{Prin}_k(\mathbf{S}) = \mathbf{\Pi}_1\mathbf{\Sigma}\mathbf{\Pi}_2$ [illegible] $\mathbf{\Pi}_1$ [illegible] $\mathbf{\Pi}_2$ [illegible] $\mathbf{\Sigma}$ [illegible].

[illegible]. Since $\mathbf{S}$ is a solution of (2.2), (2.3) holds. We multiply (2.3) by $\mathbf{L}$ to obtain

$$(\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})\mathbf{\Lambda}\mathbf{M}^{-2}\mathbf{\Lambda}\mathbf{S}^* = \mu\mathbf{S}^*\mathbf{L} = \mu\mathbf{S}^*(\mathbf{\Omega} + \mathbf{S}\mathbf{S}^*).$$

Rearranging this, we have

$$\mathbf{S}^*\mathbf{\Omega} = \mathbf{E}\mathbf{S}^*, \tag{2.10}$$

where

$$\mathbf{E} = \frac{1}{\mu}(\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})(\mathbf{\Lambda}\mathbf{M}^{-2}\mathbf{\Lambda}) - \mathbf{S}^*\mathbf{S}.$$

Since $\mathbf{\Lambda}$, $\mathbf{\Omega}$, and $\mathbf{\Theta}$ are diagonal and $\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$ is block diagonal, it follows that $\mathbf{M} = \mathbf{\Lambda}\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}\mathbf{\Lambda} + \mathbf{\Theta}$ is block diagonal and $\mathrm{Prin}_k(\mathbf{M})$ is diagonal. By (2.7), $\mathbf{S}^*\mathbf{S}$ is block diagonal and $\mathrm{Prin}_k(\mathbf{S}^*\mathbf{S})$ is diagonal. Hence, $\mathbf{E}$ is block diagonal and $\mathrm{Prin}_k(\mathbf{E})$ is diagonal. Let e_i denote the ith diagonal element of $\mathbf{E}$. For $1 \le i \le k$ and $1 \le j \le m$, we equate the (i, j) elements in (2.10) to obtain

$$(\mathbf{S}^*)_{ij}\omega_j = e_i(\mathbf{S}^*)_{ij} \quad \text{or} \quad (\mathbf{S}^*)_{ij}(\omega_j - e_i) = 0.$$

If $(\mathbf{S}^*)_{ij} \neq 0$, then $\omega_j = e_i$. By the nondegeneracy assumption, the ω_j, $1 \le j \le m$, are all distinct. Consequently, there is at most one j for which $(\mathbf{S}^*)_{ij} \neq 0$. In other words, each of the first k columns of $\mathbf{S}$ has at most one nonzero. Since $\mathrm{Prin}_k(\mathbf{S}^*\mathbf{S})$ is diagonal, no two of the leading k columns of $\mathbf{S}$ can have their single nonzero in the same row. A suitable permutation of the rows and the first k columns of $\mathbf{S}$ yields a diagonal matrix $\mathbf{\Sigma}$. □

We now apply Lemma 2.2 to the case $\mathbf{\Lambda} = \mathbf{I}$:

LEMMA 2.3. [illegible] $\mathbf{\Lambda} = \mathbf{I}$ [illegible] (2.2) [illegible] $\mathbf{S} = \mathbf{\Pi}_1\mathbf{\Sigma}\mathbf{\Pi}_2$ [illegible] $\mathbf{\Pi}_1$ [illegible] $\mathbf{\Pi}_2$ [illegible] $\mathbf{\Sigma}$ [illegible].

[illegible]. Since any $\mathbf{\Omega}$ and $\mathbf{\Theta}$ can be approximated arbitrarily closely by nondegenerate matrices, there is no loss of generality in assuming that $\mathbf{\Omega}$ and $\mathbf{\Theta}$ are nondegenerate

(see [2]). There exists an optimal solution of (2.2) since the feasible set is compact and the cost function is a continuous function of $\mathbf{S}$.

By Lemma 2.1, the matrices $\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$ and $\mathbf{\Lambda}\mathbf{M}^{-2}\mathbf{\Lambda}$ commute. Since $\mathbf{\Lambda} = \mathbf{I}$, it follows that $\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$ and $\mathbf{M}^{-2}$ commute. Since commuting matrices share a common set of eigenvectors [11, p. 249], and since the eigenvectors of $\mathbf{M}^{-2}$ and $\mathbf{M}$ are the same, it follows that $\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$ and $\mathbf{M}$ commute:

$$(\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})\mathbf{M} = \mathbf{M}(\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}).$$

This implies that

$$(\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})(\mathbf{\Theta} + \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}) = (\mathbf{\Theta} + \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})(\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}).$$

which reduces to

$$(\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})\mathbf{\Theta} = \mathbf{\Theta}(\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}).$$

Since $\mathbf{\Theta}$ satisfies the nondegeneracy condition, we conclude that $\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$ is diagonal. Taking $k = n$ (the number of columns in $\mathbf{S}$) in Lemma 2.2, $\mathbf{S} = \mathrm{Prin}_k(\mathbf{S}) = \mathbf{\Pi}_1\mathbf{\Sigma}\mathbf{\Pi}_2$. □

The case $\mathbf{\Theta} = \mathbf{I}$ and $\mathbf{\Lambda} \neq \mathbf{I}$ is tougher to analyze. In an effort to simplify the structure of a solution to (2.2), we will apply a permutation to our problem. Let $\mathbf{\Pi}$ be a permutation matrix which we will apply to the columns of $\mathbf{S}$. The permuted matrix is $\mathbf{S}_p = \mathbf{S}\mathbf{\Pi}$. Let $\mathbf{\Lambda}_p = \mathbf{\Pi}^*\mathbf{\Lambda}\mathbf{\Pi}$ be the symmetric permutation of the rows and columns of $\mathbf{\Lambda}$. In essence, $\mathbf{\Lambda}_p$ is obtained from $\mathbf{\Lambda}$ by interchanging diagonal elements. Similarly, $\mathbf{\Theta}_p = \mathbf{\Pi}^*\mathbf{\Theta}\mathbf{\Pi}$ denotes the symmetric permutation of $\mathbf{\Theta}$. We replace $\mathbf{S}$, $\mathbf{\Lambda}$, and $\mathbf{\Theta}$ by their representation in terms of the permuted quantities to obtain the following equivalent form of (2.2) (after taking into account the fact that the trace is invariant under a similarity transformation):

$$\min_{\mathbf{S}_p} \quad \mathrm{tr}\ ((\mathbf{S}_p\mathbf{\Lambda}_p)^*(\mathbf{\Omega} + \mathbf{S}_p\mathbf{S}_p^*)^{-1}\mathbf{S}_p\mathbf{\Lambda}_p + \mathbf{\Theta}_p)^{-1} \tag{2.11}$$

$$\text{subject to} \quad \mathrm{tr}\ (\mathbf{S}_p\mathbf{S}_p^*) \leq P, \quad \mathbf{S}_p \in \mathbb{C}^{m\times n}.$$

We begin with the following result:

LEMMA 2.4. $\mathbf{D}$

$$\mathbf{D} = \mathbf{S}^*\mathbf{L}^{-1/2}(\mathbf{I} + \mathbf{W}^*\mathbf{W})^{-2}\mathbf{L}^{-1/2}\mathbf{S}, \tag{2.12}$$

$\mathbf{W} = \mathbf{\Lambda}\mathbf{S}^*\mathbf{L}^{-1/2}$ $\mathbf{L} = \mathbf{\Omega} + \mathbf{S}\mathbf{S}^*$ $d_{ii} = 0$ i $\mathbf{S}$ $\mathbf{\Theta} = \mathbf{I}$ $\mathbf{S}$ (2.2) $\mathbf{\Lambda}$ $\mathbf{D}$ $\mathbf{D}_p = \mathbf{\Pi}^*\mathbf{D}\mathbf{\Pi}$ $\mathbf{D}$ $\mathbf{\Pi}$ $\mathbf{\Lambda}_p^2\mathbf{D}_p$ $\mathbf{S}_p^*\mathbf{L}^{-1}\mathbf{S}_p$ $\mathbf{\Lambda}_p^2\mathbf{D}_p$. By the definition of $\mathbf{D}$, we have

$$d_{ii} = \|(\mathbf{I} + \mathbf{W}^*\mathbf{W})^{-1}\mathbf{L}^{-1/2}\mathbf{s}_i\|^2,$$

where $\mathbf{s}_i$ is the ith column of $\mathbf{S}$. If $d_{ii} = 0$, then $\mathbf{s}_i = \mathbf{0}$.

If $\mathbf{\Theta} = \mathbf{I}$, then $\mathbf{M}$ in (2.4) has the form $\mathbf{I} + \mathbf{W}\mathbf{W}^*$ with $\mathbf{W} = \mathbf{\Lambda}\mathbf{S}^*\mathbf{L}^{-1/2}$. It follows from the matrix modification formula (2.6) that

$$\begin{aligned}\mathbf{M}^{-1}\mathbf{W} &= (\mathbf{I} - \mathbf{W}(\mathbf{I} + \mathbf{W}^*\mathbf{W})^{-1}\mathbf{W}^*)\mathbf{W} \\ &= \mathbf{W} - \mathbf{W}(\mathbf{I} + \mathbf{W}^*\mathbf{W})^{-1}\mathbf{W}^*\mathbf{W} \\ &= \mathbf{W}(\mathbf{I} + \mathbf{W}^*\mathbf{W})^{-1}.\end{aligned}$$

Hence, we have

$$\mathbf{M}^{-2}\mathbf{W} = \mathbf{W}(\mathbf{I} + \mathbf{W}^*\mathbf{W})^{-2},$$

which implies that

$$\begin{aligned} \boldsymbol{\Lambda}\mathbf{M}^{-2}\boldsymbol{\Lambda}\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S} &= \boldsymbol{\Lambda}\mathbf{M}^{-2}\mathbf{W}\mathbf{L}^{-1/2}\mathbf{S} \\ &= \boldsymbol{\Lambda}\mathbf{W}(\mathbf{I} + \mathbf{W}^*\mathbf{W})^{-2}\mathbf{L}^{-1/2}\mathbf{S} \\ &= \boldsymbol{\Lambda}^2\mathbf{S}^*\mathbf{L}^{-1/2}(\mathbf{I} + \mathbf{W}^*\mathbf{W})^{-2}\mathbf{L}^{-1/2}\mathbf{S} = \boldsymbol{\Lambda}^2\mathbf{D}, \end{aligned} \tag{2.13}$$

where $\mathbf{D}$ is defined in (2.12). By (2.13), $\boldsymbol{\Lambda}^2\mathbf{D}$ is the product of Hermitian matrices $\boldsymbol{\Lambda}\mathbf{M}^{-2}\boldsymbol{\Lambda}$ and $\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$. The matrices $\boldsymbol{\Lambda}\mathbf{M}^{-2}\boldsymbol{\Lambda}$ and $\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$ commute by Lemma 2.1. Consequently, $\boldsymbol{\Lambda}^2\mathbf{D}$ is Hermitian. Since $\mathbf{D}$ and $\boldsymbol{\Lambda}$ are also Hermitian, we have

$$(\boldsymbol{\Lambda}^2\mathbf{D}) = (\boldsymbol{\Lambda}^2\mathbf{D})^* = \mathbf{D}\boldsymbol{\Lambda}^2.$$

Since the diagonal elements of $\boldsymbol{\Lambda}$ are distinct, $\mathbf{D}$ is diagonal.

By Lemma 2.1, we can commute the factors $\mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}$ and $\boldsymbol{\Lambda}\mathbf{M}^{-2}\boldsymbol{\Lambda}$ in (2.8). Utilizing (2.13) in (2.8) gives

$$\mu\mathbf{S}^*\mathbf{S} = \boldsymbol{\Lambda}^2\mathbf{D}(\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}). \tag{2.14}$$

Inserting (2.13) in (2.7) gives

$$\mu\mathbf{S}^*\mathbf{S} = (\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})\boldsymbol{\Lambda}^2\mathbf{D}. \tag{2.15}$$

We equate the right sides of (2.14) and (2.15) to deduce that

$$(\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})\boldsymbol{\Lambda}^2\mathbf{D} = \boldsymbol{\Lambda}^2\mathbf{D}(\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S}).$$

Hence, the matrix $(\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})$ and the diagonal matrix $\boldsymbol{\Lambda}^2\mathbf{D}$ commute, and they share a common set of eigenvectors.

Suppose that $\boldsymbol{\Pi}$ is chosen so that the diagonal elements of $\boldsymbol{\Lambda}_p^2\mathbf{D}_p$ are in decreasing order. Hence, zero diagonal elements in $\mathbf{D}_p$ trail at the end of the diagonal, and the corresponding (trailing) columns of $\mathbf{S}_p$ vanish, as shown at the start of the proof. Suppose that $\lambda_i^2 d_{ii} = \lambda_j^2 d_{jj}$ for i and $j \in [p, q]$. The eigenvectors of $\boldsymbol{\Lambda}_p^2\mathbf{D}_p$ correspond to columns p through q of the identity matrix. Since $\boldsymbol{\Lambda}_p^2\mathbf{D}_p$ and $(\mathbf{I} - \mathbf{S}_p^*\mathbf{L}^{-1}\mathbf{S}_p)$ share a common set of eigenvectors, the corresponding eigenvectors of $(\mathbf{I} - \mathbf{S}_p^*\mathbf{L}^{-1}\mathbf{S}_p)$ are linear combinations of columns p through q of the identity matrix. Hence, $\mathbf{S}_p^*\mathbf{L}^{-1}\mathbf{S}_p$ is block diagonal and the size of the blocks is equal to the number of times a positive diagonal element of $\boldsymbol{\Lambda}_p^2\mathbf{D}_p$ repeats. □

Let $\boldsymbol{\Lambda}_k$ and $\boldsymbol{\Omega}_k$, $k = 1, 2, \ldots$, be nondegenerate matrices which approach limits $\boldsymbol{\Lambda}$ and $\boldsymbol{\Omega}$, respectively. Let $\mathbf{S}_k$ be a solution to (2.2) corresponding to $(\boldsymbol{\Lambda}_k, \boldsymbol{\Omega}_k)$. Such a solution exists for each k since the objective function in (2.2) is continuous when $\boldsymbol{\Omega}$ and $\boldsymbol{\Theta}$ are positive definite and the feasible set is a compact set. By suitable pruning of the sequence $(\boldsymbol{\Lambda}_k, \boldsymbol{\Omega}_k)$ if necessary, there is no loss in generality in assuming that the sequence $\mathbf{S}_k$, $k = 1, 2, \ldots$, converges to a limit $\mathbf{S}$, which is a solution of (2.2) (by the continuity of the objective function). We now show, under suitable hypothesis, that the limit $\mathbf{S}$ is a permutation of a diagonal matrix.

LEMMA 2.5. $\boldsymbol{\Theta} = \mathbf{I}$ [illegible] $\mathbf{S}$ [illegible] (2.2) [illegible] $\mathbf{S}_k$, $k = 1, 2, \ldots$, [illegible] $\boldsymbol{\Lambda}_k$ [illegible]

$\boldsymbol{\Omega}_k$, .. , ,-. -. ,-. , , , , , , $\boldsymbol{\Lambda}^2\mathbf{D}$ -. ,-. , .. , $\mathbf{S}$, , , -
$\boldsymbol{\Pi}_1\boldsymbol{\Sigma}\boldsymbol{\Pi}_2$, . $\boldsymbol{\Pi}_1$ - $\boldsymbol{\Pi}_2$, , , , , , , , $\boldsymbol{\Sigma}$- , ,-. ,

In randomly generated test problems, the "distinct diagonal" property of Lemma 2.5 was always satisfied.

, , , . Since the matrices $\boldsymbol{\Lambda}_k$ and $\boldsymbol{\Omega}_k$ are nondegenerate, the associated matrices $\mathbf{D}_k$ (see (2.12)) are diagonal. Since the $\mathbf{D}_k$ converge to $\mathbf{D}$, the limit $\mathbf{D}$ is diagonal. Since the positive diagonal elements of $\boldsymbol{\Lambda}^2\mathbf{D}$ are distinct, the associated diagonal elements of $\boldsymbol{\Lambda}_k^2\mathbf{D}_k$ are distinct for k sufficiently large. Assume that the columns of $\mathbf{S}$ and the rows of $\boldsymbol{\Lambda}$ are permuted so that the diagonal elements of the limit $\boldsymbol{\Lambda}^2\mathbf{D}$ are in decreasing order. Let p be the number of positive diagonal elements of $\mathbf{D}$. By Lemma 2.4, $\mathrm{Prin}_p(\mathbf{S}_k^*\mathbf{L}_k^{-1}\mathbf{S}_k)$ is diagonal. By Lemma 2.2, $\mathrm{Col}_p(\mathbf{S}_k) = \boldsymbol{\Pi}_{1k}\boldsymbol{\Sigma}_k\boldsymbol{\Pi}_{2k}$. Also, by Lemma 2.4, columns $l+1$ through n of $\mathbf{S}$ vanish. Hence, the limit $\mathbf{S}$ can be expressed as a product $\boldsymbol{\Pi}_1\boldsymbol{\Sigma}\boldsymbol{\Pi}_2$. □

Due to the ordering (1.5), one of the permutations in Lemma 2.3 or Lemma 2.5 can be eliminated.

THEOREM 2.6. , ... $\boldsymbol{\Theta} = \mathbf{I}$, $\boldsymbol{\Lambda} = \mathbf{I}$, (2.2) , , , , ,
$\mathbf{S} = \boldsymbol{\Pi}_1\boldsymbol{\Sigma}\boldsymbol{\Pi}_2$, . $\boldsymbol{\Sigma}$- , , , , $\boldsymbol{\Pi}_i$, , , , , (2.2)
, , , , $\mathbf{S} = \boldsymbol{\Sigma}\boldsymbol{\Pi}$, . $\boldsymbol{\Sigma}$- , , , , $\boldsymbol{\Pi}$- , , , ,
, , , , , , , $\boldsymbol{\sigma}$, , , , $\boldsymbol{\Sigma}$. p , , , , , , , ,
p- , , , , , , , $\boldsymbol{\Lambda}$, $\boldsymbol{\Pi}$, , , , , , p , , $\boldsymbol{\Sigma}$
, , , , , , (2.2) , , , $\mathbf{S} = \boldsymbol{\Pi}\boldsymbol{\Sigma}$
, , . The substitution $\mathbf{S} = \boldsymbol{\Pi}_1\boldsymbol{\Sigma}\boldsymbol{\Pi}_2$ in (2.2) yields the following equivalent problem (assuming $l = n$):

$$\min_{\boldsymbol{\Pi}_1,\boldsymbol{\Pi}_2} \ \mathrm{tr}\,\left((\boldsymbol{\Pi}_2\boldsymbol{\Lambda}\boldsymbol{\Pi}_2^*)\boldsymbol{\Sigma}^*(\boldsymbol{\Pi}_1^*\boldsymbol{\Omega}\boldsymbol{\Pi}_1 + \boldsymbol{\Sigma}\boldsymbol{\Sigma}^*)^{-1}\boldsymbol{\Sigma}(\boldsymbol{\Pi}_2\boldsymbol{\Lambda}\boldsymbol{\Pi}_2^*) + \boldsymbol{\Pi}_2\boldsymbol{\Theta}\boldsymbol{\Pi}_2^*\right)^{-1} \quad \text{subject to } \sum_{i=1}^{N}\sigma_i^2 \le P, \tag{2.16}$$

where N is the minimum of m and n. Here the minimization is over diagonal matrices $\boldsymbol{\Sigma}$ with $\boldsymbol{\sigma}$ on the diagonal, and permutation matrices $\boldsymbol{\Pi}_1$ and $\boldsymbol{\Pi}_2$.

A symmetric permutation such as $\boldsymbol{\Pi}_2\boldsymbol{\Lambda}\boldsymbol{\Pi}_2^*$ interchanges diagonal elements. Hence, (2.16) is equivalent to

$$\min_{\pi_1,\pi_2} \ \sum_{i=1}^{N}\frac{\omega_{\pi_1(i)} + \sigma_i^2}{\theta_{\pi_2(i)}\omega_{\pi_1(i)} + (\theta_{\pi_2(i)} + \lambda_{\pi_2(i)}^2)\sigma_i^2} \quad \text{subject to } \sum_{i=1}^{N}\sigma_i^2 \le P, \quad \pi_1 \in \mathcal{P}_m, \quad \pi_2 \in \mathcal{P}_n, \tag{2.17}$$

where $\mathcal{P}_m$ is the set of bijections of $\{1, 2, \ldots, m\}$ onto itself.

Let $\boldsymbol{\sigma}$ denote an optimal solution of (2.17). If $\lambda_{\pi_2(i)} = 0$, then the associated term in the objective function of (2.17) reduces to $1/\theta_{\pi_2(i)}$, independent of σ_i. In this case $\sigma_i = 0$ is optimal (see Theorem 3.1 and (3.2) in the next section). Hence, the number of positive components of $\boldsymbol{\sigma}$ is less than or equal to the rank of $\boldsymbol{\Lambda}$. Define the set

$$\mathcal{S} = \{i : \sigma_i > 0\},$$

and let $p = |\mathcal{S}|$. The function

$$\frac{\omega + x}{\omega\theta + (\theta + \lambda^2)x}, \quad x > 0$$

is monotone increasing in $\omega \geq 0$ and monotone decreasing in $\lambda \geq 0$. Since the objective function is being minimized in (2.17), it follows that $\omega_{\pi_1(i)}$ is one of the p smallest elements of $\boldsymbol{\omega}$. In the same fashion, if $\boldsymbol{\Theta} = \mathbf{I}$ and $i \in \mathcal{S}$, then $\lambda_{\pi_2(i)}$ is one of the p largest elements of $\boldsymbol{\lambda}$.

Finally, let us consider the case $\boldsymbol{\Lambda} = \mathbf{I}$. The cost function in (2.17) is the sum of two expressions:

$$\sum_{i\in\mathcal{S}} \frac{\omega_{\pi_1(i)} + \sigma_i^2}{\theta_{\pi_2(i)}\omega_{\pi_1(i)} + (\theta_{\pi_2(i)} + \lambda^2_{\pi_2(i)})\sigma_i^2} \quad + \quad \sum_{j\in\mathcal{S}^c} \frac{1}{\theta_{\pi_2(j)}}. \tag{2.18}$$

We now show that if $i \in \mathcal{S}$, but $\theta_{\pi_2(i)}$ is not one of the p smallest elements of $\boldsymbol{\theta}$, then the cost function is decreased by exchanging $\pi_2(i)$ with $\pi_2(j)$ where $j \in \mathcal{S}^c$ and $\theta_{\pi_2(j)} < \theta_{\pi_2(i)}$. Let $\beta_1 = \theta_{\pi_2(i)}$ and $\beta_2 = \theta_{\pi_2(j)}$, and define

$$V_1 = \frac{\omega + \sigma^2}{\omega\beta_1 + \sigma^2(\beta_1 + 1)} + \frac{1}{\beta_2} \quad \text{and} \quad V_2 = \frac{\omega + \sigma^2}{\omega\beta_2 + \sigma^2(\beta_2 + 1)} + \frac{1}{\beta_1}.$$

Here V_1 represents the i and j terms in (2.18) after substituting $\lambda_{\pi_2(i)} = 1$, while V_2 reflects the corresponding terms after the exchange of $\pi_2(i)$ with $\pi_2(j)$. Since $\beta_1 > \beta_2$, it can be shown that $V_1 - V_2 \geq 0$ (cross multiply and cancel terms). Hence, by exchanging $\pi_2(j)$ with $\pi_2(i)$, the cost function is decreased. In summary, if either $\boldsymbol{\Theta} = \mathbf{I}$ or $\boldsymbol{\Lambda} = \mathbf{I}$, then for $i \in \mathcal{S}$, $\lambda_{\pi_2}(i)$ is one of the p largest elements in $\boldsymbol{\lambda}$ while $\theta_{\pi_2(i)}$ and $\omega_{\pi_1(i)}$ are among the p smallest elements in $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$, respectively. Due to the ordering (1.5),

$$\{\pi_1(i) : i \in \mathcal{S}\} \subset \{1, 2, \ldots, p\} \quad \text{and} \quad \{\pi_2(i) : i \in \mathcal{S}\} \subset \{1, 2, \ldots, p\}.$$

Let $\pi_3 \in \mathcal{P}_N$ be chosen so that

$$\sigma_{\pi_3(1)} \geq \sigma_{\pi_3(2)} \geq \ldots \geq \sigma_{\pi_3(N)}.$$

Since $\mathcal{S}$ is the set of indices of positive components of $\boldsymbol{\sigma}$, we have

$$\mathcal{S} = \{\pi_3(i) : i = 1, 2, \ldots, p\}.$$

Define $\hat{\pi}_1 = \pi_1(\pi_3)$, $\hat{\pi}_2 = \pi_2(\pi_3)$, and $\hat{\sigma}_i = \sigma_{\pi_3(i)}$. The optimal cost (2.18) can be written

$$\sum_{i=1}^{p} \frac{\omega_{\hat{\pi}_1(i)} + \hat{\sigma}_i^2}{\theta_{\hat{\pi}_2(i)}\omega_{\hat{\pi}_1(i)} + (\theta_{\hat{\pi}_2(i)} + \lambda^2_{\hat{\pi}_2(i)})\hat{\sigma}_i^2} \quad + \quad \sum_{i>p} \frac{1}{\theta_{\hat{\pi}_2(i)}}.$$

Hence, (2.2) has a solution of the form $\widehat{\mathbf{S}} = \widehat{\boldsymbol{\Pi}}_1\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\Pi}}_2$ where $\widehat{\boldsymbol{\Pi}}_1$ permutes only the first p rows and $\widehat{\boldsymbol{\Pi}}_2$ permutes only the first p columns of $\widehat{\boldsymbol{\Sigma}}$. Let $\overline{\boldsymbol{\Pi}}_1$ be a permutation matrix which is the same as $\widehat{\boldsymbol{\Pi}}_1$ except that it has been expanded (by an identity matrix) or chopped ($\overline{\boldsymbol{\Pi}}_1 = \mathrm{Prin}_n(\widehat{\boldsymbol{\Pi}}_1)$) to match the number of columns of $\mathbf{S}$. Define $\boldsymbol{\Sigma}' = \widehat{\boldsymbol{\Pi}}_1\widehat{\boldsymbol{\Sigma}}\overline{\boldsymbol{\Pi}}_1^*$. $\boldsymbol{\Sigma}'$ is diagonal since it is a symmetric permutation of a diagonal matrix. Consequently, we have

$$\widehat{\mathbf{S}} = \widehat{\boldsymbol{\Pi}}_1\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\Pi}}_2 = \widehat{\boldsymbol{\Pi}}_1\widehat{\boldsymbol{\Sigma}}\overline{\boldsymbol{\Pi}}_1^*\overline{\boldsymbol{\Pi}}_1\widehat{\boldsymbol{\Pi}}_2 = \boldsymbol{\Sigma}'\boldsymbol{\Pi},$$

where $\boldsymbol{\Pi} = \overline{\boldsymbol{\Pi}}_1\widehat{\boldsymbol{\Pi}}_2$. In a similar manner, we obtain $\mathbf{S} = \boldsymbol{\Pi}\boldsymbol{\Sigma}$ for a different choice of $\boldsymbol{\Pi}$ and $\boldsymbol{\Sigma}$. □

Based on Theorem 2.6, one of the permutation in (2.17) can be deleted when $\mathbf{\Lambda} = \mathbf{I}$ or $\mathbf{\Theta} = \mathbf{I}$. We delete the permutation π_2 to obtain the following problem:

$$\min_{\mathbf{s},\pi} \sum_{i=1}^{M} \frac{\omega_{\pi(i)} + s_i}{\theta_i \omega_{\pi(i)} + (\theta_i + \lambda_i) s_i} \quad \text{subject to} \quad \sum_{i=1}^{M} s_i \le P, \quad \mathbf{s} \ge \mathbf{0}, \quad \pi \in \mathcal{P}_M, \tag{2.19}$$

where $s_i = \sigma_i^2$ and M is the minimum of m and the rank of $\mathbf{\Lambda}$. If $\mathbf{s}$ and π are solutions of (2.19), then $\mathbf{S} = \mathbf{\Pi\Sigma}$ where $\sigma_i^2 = s_i$ is a solution of (2.2). We now combine Theorem 2.6 with the change of variables (2.1).

COROLLARY 2.7. , $\mathbf{H} = \mathbf{I}$, (1.4) , $\mathbf{P} = \mathbf{V}_w \mathbf{\Sigma\Pi V}_t^*$, $\mathbf{\Pi}$, $\mathbf{\Sigma}$, $\mathbf{\Theta} = \mathbf{I}$ (2.2) , 2.5 , (1.4) , $\mathbf{P} = \mathbf{V}_w \mathbf{\Sigma\Pi U}^*$

As in Theorem 2.6, the factor $\mathbf{\Sigma\Pi}$ in Corollary 2.7 can be replaced by $\mathbf{\Pi\Sigma}$.

3. The optimal $\mathbf{\Sigma}$. Assuming the permutation π in (2.19) is given, let us now consider the problem of optimizing over $\boldsymbol{\sigma}$. To simplify the indexing, the permutation is suppressed and we consider the problem:

$$\min \sum_{i=1}^{M} \frac{\omega_i + s_i}{\theta_i \omega_i + (\theta_i + \lambda_i^2) s_i} \quad \text{subject to} \quad \sum_{i=1}^{M} s_i \le P, \quad \mathbf{s} \ge \mathbf{0}. \tag{3.1}$$

The solution of (3.1) can be expressed in terms of a Lagrange multiplier for the constraint (this solution technique is often called "water filling" [1] in the communication literature).

THEOREM 3.1. , (3.1) ,

$$s_i = \frac{1}{\theta_i + \lambda_i^2} \max\left\{ \sqrt{\frac{\omega_i \lambda_i^2}{\mu}} - \theta_i \omega_i, \ 0 \right\}, \tag{3.2}$$

, μ ,

$$\sum_{i=1}^{M} s_i = P. \tag{3.3}$$

. Since the minimization in (3.1) takes place over a closed, bounded set, there exists a solution. Since the function $(\omega_i + x)/(\theta_i \omega_i + (1 + \lambda_i^2)x)$ is a decreasing function of $x \ge 0$, the objective function decreases when s_i increases. Hence, there exists a solution of (3.1) with the inequality constraint active. Due to the strict convexity of the cost function and the convexity of the constraints, (3.1) has a unique solution.

The first-order optimality conditions (KKT conditions) for an optimal solution of (3.1) are the following: There exists a scalar $\mu \ge 0$ and a vector $\boldsymbol{\nu} \in \mathbb{R}^M$ such that

$$\mu - \nu_i - \frac{\omega_i \lambda_i^2}{(\theta_i \omega_i + (\theta_i + \lambda_i^2) s_i)^2} = 0, \quad \nu_i \ge 0, \quad s_i \ge 0, \quad \text{and} \quad \nu_i s_i = 0, \tag{3.4}$$

$1 \le i \le M$. Any solution of (3.4) is the unique optimal solution of (3.1).

A solution to (3.4) is obtained as follows: Define the function

$$s_i(\mu) = \frac{1}{\theta_i + \lambda_i^2}\left(\sqrt{\frac{\omega_i \lambda_i^2}{\mu}} - \theta_i \omega_i\right)^+. \tag{3.5}$$

Here $x^+ = \max\{x, 0\}$. This particular value for s_i is obtained by setting $\nu_i = 0$ in (3.4), solving for s_i, and replacing the solution by 0 when it is negative. Observe that $s_i(\mu)$ is a decreasing function of μ that approaches $+\infty$ as μ approaches 0 and that approaches 0 as μ tends to $+\infty$. Hence, the equation

$$\sum_{i=1}^{M} s_i(\mu) = P \tag{3.6}$$

has a unique positive solution. Observe that $s_i(\mu) = 0$ if and only if $\mu \geq \lambda_i^2/(\theta_i^2 \omega_i)$. Moreover, if $\mu \geq \lambda_i^2/(\omega_i \theta_i^2)$, then

$$\mu - \frac{\omega_i \lambda_i^2}{(\theta_i \omega_i + (\theta_i + \lambda_i^2) s_i(\mu))^2} = \mu - \frac{\lambda_i^2}{\omega_i \theta_i^2} \geq 0.$$

It follows that the KKT conditions are satisfied by the positive solution of (3.6). ∎

4. Optimal permutation for $\mathbf{\Lambda} = \mathbf{I}$. Starting with this section, we will determine optimal permutations π in (2.19). When $\mathbf{\Lambda} = \mathbf{I}$, the optimal permutation is the identity (due to the ordering (1.5)):

THEOREM 4.1. [illegible] $\mathbf{\Lambda} = \mathbf{I}$ [illegible] $\pi(i) = i$ [illegible] (2.19) [illegible]. Recall that the components of $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ are in increasing order. Let p be the number of positive components of an optimal $\mathbf{s}$ in (2.19). By Theorem 2.6, an optimal permutation π permutes only the first p components of $\boldsymbol{\omega}$; moreover, $s_i > 0$ for $i \leq p$ and $s_i = 0$ for $i > p$.

Suppose that there exists a permutation π which is optimal in (2.19) and with the property that $\omega_{\pi(i)} > \omega_{\pi(j)}$ for some $i < j \leq p$. Since the components of $\boldsymbol{\theta}$ are in increasing order, $\theta_i \leq \theta_j$. We will show that by interchanging components i and j of π, the objective function value does not increase. Consequently, after a finite number of pairwise exchanges, and without increasing the cost, it can be arranged so that $\omega_{\pi(i)}$ is an increasing function of i. Since $1 \leq \pi(i) \leq p$ for $i \leq p$ and since the components of $\boldsymbol{\omega}$ are in increasing order, we conclude that $\pi(i) = i$ for all i is optimal in (2.19).

Let $\mathbf{s}$ denote a solution of (2.19) associated with the permutation π and suppose that $\omega_{\pi(i)} > \omega_{\pi(j)}$ for some $i < j \leq p$. For notational convenience, let us take $i = 1$, $j = 2$, $\pi(1) = 2$, and $\pi(2) = 1$. Define $\omega_1' = \omega_{\pi(1)} = \omega_2$ and $\omega_2' = \omega_{\pi(2)} = \omega_1$. Due to the optimality of $\mathbf{s}$ and π, $t_1 = s_1 > 0$ and $t_2 = s_2 > 0$ is an optimal solution of the following 2-variable problem:

$$\min_{\mathbf{t}} \sum_{i=1}^{2} \frac{\omega_i' + t_i}{\theta_i \omega_i' + (\theta_i + 1) t_i} \tag{4.1}$$

$$\text{subject to} \quad \sum_{i=1}^{2} t_i \leq \bar{P} := s_1 + s_2, \quad \mathbf{t} \geq \mathbf{0}.$$

We will show that the optimal objective function value for the following unpermuted problem is less than or equal to the objective function value for (4.1):

$$\min_{\mathbf{t}} \sum_{i=1}^{2} \frac{\omega_i + t_i}{\theta_i \omega_i + (\theta_i + 1)t_i} \quad \text{subject to} \quad \sum_{i=1}^{2} t_i \le \bar{P}, \quad \mathbf{t} \ge \mathbf{0}. \tag{4.2}$$

By assumption, the solution of (4.1) is strictly positive. We now show that this implies the solution of (4.2) is strictly positive. By Theorem 3.1, the condition $\mathbf{s} > \mathbf{0}$ is equivalent to

$$1/\sqrt{\mu} > \theta_i \sqrt{\omega_i'}. \tag{4.3}$$

The multiplier μ is given by

$$\frac{1}{\sqrt{\mu}} = \frac{\bar{P} + \sum_{j=1}^{2} \frac{\theta_j \omega_j'}{(\theta_j + 1)}}{\sum_{j=1}^{2} \frac{{\omega_j'}^{1/2}}{(\theta_j + 1)}}. \tag{4.4}$$

Combining (4.3) and (4.4) gives

$$\bar{P} > \frac{\theta_1 \sqrt{\omega_1' \omega_2'}}{(1+\theta_2)} - \frac{\theta_2 \omega_1'}{(1+\theta_2)} = \frac{\theta_1 \sqrt{\omega_1 \omega_2}}{(1+\theta_2)} - \frac{\theta_2 \omega_2}{(1+\theta_2)} \text{ and} \tag{4.5}$$

$$\bar{P} > \frac{\theta_2 \sqrt{\omega_1' \omega_2'}}{(1+\theta_1)} - \frac{\theta_1 \omega_2'}{(1+\theta_1)} = \frac{\theta_2 \sqrt{\omega_1 \omega_2}}{(1+\theta_1)} - \frac{\theta_1 \omega_1}{(1+\theta_1)}. \tag{4.6}$$

Above, the first inequality corresponds to the condition $s_1 > 0$ while the second corresponds to $s_2 > 0$. Similarly, the optimal $\mathbf{t}$ in (4.2) is positive if and only if

$$\bar{P} > \frac{\theta_1 \sqrt{\omega_1 \omega_2}}{(1+\theta_2)} - \frac{\theta_2 \omega_1}{(1+\theta_2)} \text{ and} \tag{4.7}$$

$$\bar{P} > \frac{\theta_2 \sqrt{\omega_1 \omega_2}}{(1+\theta_1)} - \frac{\theta_1 \omega_2}{(1+\theta_1)}. \tag{4.8}$$

Since $\omega_1 \le \omega_2$, (4.6) implies that (4.8) holds. Since $\theta_1 \le \theta_2$, we have

$$\frac{1}{1+\theta_1} \ge \frac{1}{1+\theta_2} \quad \text{and} \quad \frac{\theta_1}{1+\theta_1} \le \frac{\theta_2}{1+\theta_2}. \tag{4.9}$$

Combining this with (4.6) gives

$$\begin{aligned} \bar{P} &> \frac{\theta_2 \sqrt{\omega_1 \omega_2}}{(1+\theta_1)} - \frac{\theta_1 \omega_1}{(1+\theta_1)} \ge \frac{\theta_2 \sqrt{\omega_1 \omega_2}}{(1+\theta_2)} - \frac{\theta_2 \omega_1}{(1+\theta_2)} \\ &\ge \frac{\theta_1 \sqrt{\omega_1 \omega_2}}{(1+\theta_2)} - \frac{\theta_2 \omega_1}{(1+\theta_2)}. \end{aligned}$$

Hence, (4.7) is satisfied. Since both (4.7) and (4.8) are satisfied, it follows that the solution to (4.2) is strictly positive.

Using the solution given by Theorem 3.1 and the multiplier (4.4), we obtain the following expression for the optimal objective function value C' for (4.1) (the algebra is omitted):

$$\begin{aligned} C' &= \frac{1}{1+\theta_1} + \frac{1}{1+\theta_2} + \frac{\left(\dfrac{\sqrt{\omega_1'}}{(1+\theta_1)} + \dfrac{\sqrt{\omega_2'}}{(1+\theta_2)}\right)^2}{\bar{P} + \dfrac{\theta_1\omega_1'}{(1+\theta_1)} + \dfrac{\theta_2\omega_2'}{(1+\theta_2)}} \\ &= \frac{1}{1+\theta_1} + \frac{1}{1+\theta_2} + \frac{\left(\dfrac{\sqrt{\omega_2}}{(1+\theta_1)} + \dfrac{\sqrt{\omega_1}}{(1+\theta_2)}\right)^2}{\bar{P} + \dfrac{\theta_1\omega_2}{(1+\theta_1)} + \dfrac{\theta_2\omega_1}{(1+\theta_2)}}. \end{aligned} \tag{4.10}$$

Similarly, the optimal objective function value C for (4.2) is obtained by erasing the primes in (4.10):

$$C = \frac{1}{1+\theta_1} + \frac{1}{1+\theta_2} + \frac{\left(\dfrac{\sqrt{\omega_1}}{(1+\theta_1)} + \dfrac{\sqrt{\omega_2}}{(1+\theta_2)}\right)^2}{\bar{P} + \dfrac{\theta_1\omega_1}{(1+\theta_1)} + \dfrac{\theta_2\omega_2}{(1+\theta_2)}}. \tag{4.11}$$

We will show that $C \le C'$.

Recall the following majorization property [7, p. 141]: If $\mathbf{a}$ and $\mathbf{b} \in \mathbb{R}^n$, then

$$\sum_{i=1}^n a_{[i]} b_{[n-i+1]} \le \sum_{i=1}^n a_i b_i \le \sum_{i=1}^n a_{[i]} b_{[i]},$$

where $a_{[i]}$ denotes the ith largest component of $\mathbf{a}$. We apply the inequality (4.9) and $\omega_1 \le \omega_2$ and the majorization property to the numerators in (4.10) and (4.11) to obtain

$$\left(\frac{\sqrt{\omega_1}}{1+\theta_1} + \frac{\sqrt{\omega_2}}{1+\theta_2}\right)^2 \le \left(\frac{\sqrt{\omega_2}}{1+\theta_1} + \frac{\sqrt{\omega_1}}{1+\theta_2}\right)^2 = \left(\frac{\sqrt{\omega_1'}}{1+\theta_1} + \frac{\sqrt{\omega_2'}}{1+\theta_2}\right)^2.$$

Also, by (4.9) and the majorization property, the denominators in (4.10) and (4.11) satisfy

$$\left(\frac{\theta_1\omega_1}{1+\theta_1} + \frac{\theta_2\omega_2}{1+\theta_2}\right) \ge \left(\frac{\theta_1\omega_2}{1+\theta_1} + \frac{\theta_2\omega_1}{1+\theta_2}\right).$$

Hence, $C \le C'$. This completes the proof. □

5. Optimal permutation for $\boldsymbol{\Theta} = \mathbf{I}$ and large power. When $\boldsymbol{\Theta} = \mathbf{I}$, the optimal permutation depends on P. In this section, we determine the optimal permutation when P is large, while the next section analyzes the case of small P. As shown in Theorem 2.6, the solution to (2.2) can written as either $\boldsymbol{\Pi\Sigma}$ or $\boldsymbol{\Sigma\Pi}$. When

$\boldsymbol{\Theta} = \mathbf{I}$, the analysis is simpler when we take $\mathbf{S} = \boldsymbol{\Sigma}\boldsymbol{\Pi}$, in which case (2.2) reduces to (see (2.17))

$$\min_{\mathbf{s},\pi} \sum_{i=1}^{M} \frac{\omega_i + s_i}{\omega_i + (1+\lambda_{\pi(i)}^2)s_i} \tag{5.1}$$

$$\text{subject to} \quad \sum_{i=1}^{M} s_i \le P, \quad \mathbf{s} \ge \mathbf{0}, \quad \pi \in \mathcal{P}_M,$$

where M is the minimum of m and the rank of $\boldsymbol{\Lambda}$.

THEOREM 5.1. [illegible] π [illegible] (5.1) [illegible]

$$\frac{\lambda_{\pi(1)}}{1+\lambda_{\pi(1)}^2} \ge \frac{\lambda_{\pi(2)}}{1+\lambda_{\pi(2)}^2} \ge \cdots \ge \frac{\lambda_{\pi(M)}}{1+\lambda_{\pi(M)}^2}. \tag{5.2}$$

In the noise term $\boldsymbol{\eta}$ vanishes, the optimal permutation arranged the singular values in increasing order for large P. Since the function $\lambda/(1+\lambda^2)$ is monotone increasing for $\lambda \in [0,1]$ and monotone decreasing for $\lambda > 1$, it follows that when $\boldsymbol{\eta}$ is included in the model and when its covariance is $\mathbf{I}$, the singular values smaller than one are in decreasing order, while the singular values larger than one are in increasing order. Hence, when $\boldsymbol{\Theta} = \mathbf{I}$, the solution of the problem when $\boldsymbol{\eta}$ is included in the model is fundamentally different from the solution when the noise $\boldsymbol{\eta}$ is neglected.

[illegible]. Referring to Theorem 3.1, as P tends to infinity, the optimal multiplier μ tends to zero; consequently, as P tends to infinity, all the components of the optimal $\mathbf{s}$ tend to infinity. We assume that P is large enough that for any permutation of the components of $\boldsymbol{\lambda}$, the $\mathbf{s}$ that satisfies (3.2) and (3.3) is strictly positive.

So far, we have assumed that the components of $\boldsymbol{\lambda}$ are in decreasing order (1.5). In the proof of this theorem, it is more convenient to assume that the components of $\boldsymbol{\lambda}$ are arranged in the order (5.2). In other words,

$$\frac{\lambda_1}{1+\lambda_1^2} \ge \frac{\lambda_2}{1+\lambda_2^2} \ge \cdots \ge \frac{\lambda_M}{1+\lambda_M^2}.$$

Let π by an optimal permutation in (5.1) and define $\lambda_i' = \lambda_{\pi(i)}$. Suppose for some $i < j$, we have $\lambda_i'/(1+\lambda_i'^2) < \lambda_j'/(1+\lambda_j'^2)$. We will show that by interchanging the values of $\pi(i)$ and $\pi(j)$, the objective function cannot increase. Hence, after a finite series of pairwise exchanges, we obtain (5.2) without increasing the objective function.

As in Theorem 4.1, we assume for notational convenience that $i = 1$, $j = 2$, $\pi(1) = 2$, and $\pi(2) = 1$. To summarize, we have

$$\omega_1 \le \omega_2, \quad \lambda_1' = \lambda_2, \quad \lambda_2' = \lambda_1, \quad \text{and} \quad \frac{\lambda_1}{1+\lambda_1^2} > \frac{\lambda_2}{1+\lambda_2^2}. \tag{5.3}$$

If $\mathbf{s}$ is a solution of (5.1), then $t_1 = s_1$ and $t_2 = s_2$ is a solution to

$$\min_{,\pi} \sum_{i=1}^{2} \frac{\omega_i + t_i}{\omega_i + (1+\lambda_i'^2)t_i} \tag{5.4}$$

$$\text{subject to} \quad \sum_{i=1}^{2} t_i \le \bar{P} := s_1 + s_2, \quad \mathbf{t} \ge \mathbf{0}.$$

The unpermuted problem is obtained by erasing the prime:

$$\min_{,\pi} \sum_{i=1}^{2} \frac{\omega_i + t_i}{\omega_i + (1+\lambda_i{}^2)t_i} \quad \text{subject to} \quad \sum_{i=1}^{2} t_i \le \bar{P}, \quad \mathbf{t} \ge \mathbf{0}. \tag{5.5}$$

If $\omega_1 = \omega_2$, then the optimal cost C' for the permuted problem (5.4) equals the optimal cost C for the unpermuted problem since the objective functions are identical. Hence, by interchanging the values of $\pi(1)$ and $\pi(2)$, the objective function value does not change.

Now, let us consider the case where $\omega_1 < \omega_2$. We define

$$N' = \frac{\lambda_1'\sqrt{\omega_1}}{1+\lambda_1'{}^2} + \frac{\lambda_2'\sqrt{\omega_2}}{1+\lambda_2'{}^2} \quad \text{and} \quad D' = \frac{\omega_1}{1+\lambda_1'{}^2} + \frac{\omega_2}{1+\lambda_2'{}^2} = \frac{\omega_1}{1+\lambda_2{}^2} + \frac{\omega_2}{1+\lambda_1{}^2}.$$

Parameters N and D are obtained by erasing the primes in N' and D'. With this notation the multiplier μ given by Theorem 3.1 for the problem (5.4) can be expressed

$$\frac{1}{\sqrt{\mu}} = \frac{\bar{P} + D'}{N'}. \tag{5.6}$$

Moreover, the optimal objective function value C' for (5.4) is

$$C' = \frac{1}{1+\lambda_1'{}^2} + \frac{1}{1+\lambda_2'{}^2} + \frac{N'^2}{\bar{P}+D'} = \frac{1}{1+\lambda_1{}^2} + \frac{1}{1+\lambda_2{}^2} + \frac{N'^2}{\bar{P}+D'}. \tag{5.7}$$

Similarly, the optimal objective function value C for the unpermuted problem is obtained by erasing the primes:

$$C = \frac{1}{1+\lambda_1{}^2} + \frac{1}{1+\lambda_2{}^2} + \frac{N^2}{\bar{P}+D}.$$

The inequality $C < C'$ is equivalent to

$$N^2(\bar{P}+D') < N'^2(\bar{P}+D).$$

Rearranging this, we have

$$\bar{P}(N+N')(N-N') = \bar{P}(N^2 - N'^2) < N'^2 D - N^2 D'.$$

Since $N + N' > 0$, it follows that

$$N - N' \le \frac{N'^2 D - N^2 D'}{\bar{P}(N+N')}. \tag{5.8}$$

By the definitions of N and N', we obtain

$$N - N' = (\sqrt{\omega_1} - \sqrt{\omega_2})\left(\frac{\lambda_1}{1+\lambda_1^2} - \frac{\lambda_2}{1+\lambda_2^2}\right) < 0$$

since $\omega_1 < \omega_2$ and (5.3) holds. Since $N - N' < 0$, it follows that (5.8) holds for P sufficiently large. Equivalently, for P sufficiently large, $C < C'$. This completes the proof. ∎

6. Optimal permutation for $\Theta = \mathbf{I}$ and small power. We now evaluate the optimal solution to (5.1) when P is small.

THEOREM 6.1. [illegible] L [illegible] γ_1 [illegible] λ_1 [illegible] ϵ [illegible]

$$\epsilon = \min\left\{\left|\frac{\sqrt{\omega_k}(\lambda_i\sqrt{\omega_l} - \lambda_j\sqrt{\omega_k})}{(1+\lambda_i^2)\lambda_j}\right| : \ i,j,k,l \in [1,M], \ \ \lambda_i\sqrt{\omega_l} \neq \lambda_j\sqrt{\omega_k}\right\}.$$

[illegible] $P < \epsilon$ [illegible] (5.1)[illegible]

$$s_i = P/L, \quad 1 \le i \le L, \quad s_i = 0, \quad i > L, \quad \pi(i) = i \text{ [illegible] } i. \tag{6.1}$$

[illegible]. Let π and $\mathbf{s}$ be optimal in (5.1) and define $\lambda'_i = \lambda_{\pi(i)}$. We now show that if $s_i > 0$, $s_j > 0$, and $P < \epsilon$, then we have $\lambda'_i\sqrt{\omega_j} = \lambda'_j\sqrt{\omega_i}$. To simplify the notation, we take $i = 1$ and $j = 2$, but in general, i and j are distinct integers between 1 and M. Since $\mathbf{s}$ yields an optimal solution of (5.1), it follows that an optimal solution for the following reduced problem is $t_1 = s_1$ and $t_2 = s_2$:

$$\begin{aligned} &\min_{t_1,t_2} \ \ \frac{\omega_1 + t_1}{\omega_1 + (1+{\lambda'_1}^2)t_1} + \frac{\omega_2 + t_2}{\omega_2 + (1+{\lambda'_2}^2)t_2} \\ &\text{subject to} \ \ t_1 + t_2 = \bar{P} := s_1 + s_2, \quad \mathbf{t} \ge \mathbf{0}. \end{aligned} \tag{6.2}$$

By Theorem 3.1, the t_i can be expressed:

$$t_i = \frac{1}{1+{\lambda'_i}^2}\left(\lambda'_i\sqrt{\frac{\omega_i}{\mu}} - \omega_i\right), \tag{6.3}$$

where μ is obtained from the condition $t_1 + t_2 = \bar{P}$:

$$\mu = \left(\frac{\displaystyle\sum_{i=1}^{2}\frac{\lambda'_i\sqrt{\omega_i}}{1+{\lambda'_i}^2}}{\bar{P} + \displaystyle\sum_{i=1}^{2}\frac{\omega_i}{1+{\lambda'_i}^2}}\right)^2.$$

By (6.3), $t_i > 0$ is equivalent to

$${\lambda'_i}^2 > \omega_i\mu = \omega_i\left(\frac{\displaystyle\sum_{i=1}^{2}\frac{\lambda'_i\sqrt{\omega_i}}{1+{\lambda'_i}^2}}{\bar{P} + \displaystyle\sum_{i=1}^{2}\frac{\omega_i}{1+{\lambda'_i}^2}}\right)^2.$$

We rearrange this to obtain

$$\lambda'_i\left(\bar{P} + \frac{\omega_1}{1+{\lambda'_1}^2} + \frac{\omega_2}{1+{\lambda'_2}^2}\right) > \sqrt{\omega_i}\left(\frac{\lambda'_1\sqrt{\omega_1}}{1+{\lambda'_1}^2} + \frac{\lambda'_2\sqrt{\omega_2}}{1+{\lambda'_2}^2}\right),$$

which reduces to

$$\bar{P}\lambda'_i > \sqrt{\omega_i}\left(\frac{\lambda'_1\sqrt{\omega_1}}{1+{\lambda'_1}^2} + \frac{\lambda'_2\sqrt{\omega_2}}{1+{\lambda'_2}^2}\right) - \lambda'_i\left(\frac{\omega_1}{1+{\lambda'_1}^2} + \frac{\omega_2}{1+{\lambda'_2}^2}\right).$$

Setting $i = 1$ and $i = 2$, respectively, we get

$$\bar{P} > \frac{\sqrt{\omega_2}(\lambda_2'\sqrt{\omega_1} - \lambda_1'\sqrt{\omega_2})}{(1+{\lambda_2'}^2)\lambda_1'}$$

and

$$\bar{P} > \frac{\sqrt{\omega_1}(\lambda_1'\sqrt{\omega_2} - \lambda_2'\sqrt{\omega_1})}{(1+{\lambda_1'}^2)\lambda_2'}.$$

Unless $\lambda_1'\sqrt{\omega_2} = \lambda_2'\sqrt{\omega_1}$, the condition $\epsilon \geq P \geq \bar{P}$ is violated. Hence, $\lambda_1'\sqrt{\omega_2} = \lambda_2'\sqrt{\omega_1}$, and in general, $\lambda_i'\sqrt{\omega_j} = \lambda_j'\sqrt{\omega_i}$ for each i and j with $s_i > 0$ and $s_j > 0$.

By the ordering (1.5), we have $\omega_1 \leq \omega_2$. If $\omega_1 < \omega_2$, then we will show that by exchanging $\pi(1)$ and $\pi(2)$ in (5.1), the value of the objective function is strictly decreased, which violates the optimality of π. In general, whenever $s_i > 0$ and $s_j > 0$, we have $\omega_i = \omega_j$. Since $\lambda_i'\sqrt{\omega_j} = \lambda_j'\sqrt{\omega_i}$ for each i and j for which $s_i > 0$ and $s_j > 0$, it follows that $\lambda_i' = \lambda_j'$. By Theorem 2.6, if $\mathbf{s}$ has p positive components, then π permutes only the p largest components of $\boldsymbol{\lambda}$. Since the components of $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$ associated with the positive components of $\mathbf{s}$ are all equal, we conclude that the positive components of $\mathbf{s}$ correspond to the minimum of the multiplicities of λ_1 and ω_1, and $\pi(i) = i$ for all i is optimal. Since the L largest components of $\boldsymbol{\lambda}$ and the L smallest components of $\boldsymbol{\omega}$ are all equal, it follows from Theorem 3.1 that the first L components of $\mathbf{s}$ are all equal. Since the s_i sum to P, $s_i = P/L$ for $1 \leq i \leq L$ and $s_i = 0$ for $i > L$, which completes the proof.

Now, let us prove that when $\omega_1 < \omega_2$, the exchange of $\pi(1)$ and $\pi(2)$ yields a strictly smaller value of the objective function, violating the optimality of π (hence, $\omega_1 = \omega_2$). By (5.7) the optimal objective function value C' for (6.2) is

$$C' = \frac{\left(\dfrac{\lambda_1'\sqrt{\omega_1}}{1+{\lambda_1'}^2} + \dfrac{\lambda_2'\sqrt{\omega_2}}{1+{\lambda_2'}^2}\right)^2}{\bar{P} + \dfrac{\omega_1}{1+{\lambda_1'}^2} + \dfrac{\omega_2}{1+{\lambda_2'}^2}} + \frac{1}{1+{\lambda_1'}^2} + \frac{1}{1+{\lambda_2'}^2}. \tag{6.4}$$

Since $\lambda_1'\sqrt{\omega_2} = \lambda_2'\sqrt{\omega_1}$, (6.4) can be written

$$\begin{aligned} C' &= \frac{\omega_1{\lambda_1'}^2\left(\dfrac{1}{1+{\lambda_1'}^2} + \dfrac{{\lambda_2'}^2}{{\lambda_1'}^2(1+{\lambda_2'}^2)}\right)^2}{\bar{P} + \omega_1\left(\dfrac{1}{1+{\lambda_1'}^2} + \dfrac{{\lambda_2'}^2}{{\lambda_1'}^2(1+{\lambda_2'}^2)}\right)} + \frac{1}{1+{\lambda_1'}^2} + \frac{1}{1+{\lambda_2'}^2} \\ &= \frac{\omega_1{\lambda_1'}^2x^2}{\bar{P}+\omega_1 x} + \frac{1}{1+{\lambda_1'}^2} + \frac{1}{1+{\lambda_2'}^2} \\ &= {\lambda_1'}^2x - \frac{\bar{P}{\lambda_1'}^2x}{\bar{P}+\omega_1 x} + \frac{1}{1+{\lambda_1'}^2} + \frac{1}{1+{\lambda_2'}^2} \\ &= {\lambda_1'}^2x - \frac{\bar{P}{\lambda_1'}^2}{\omega_1} + \frac{\bar{P}^2{\lambda_1'}^2}{\omega_1(\bar{P}+\omega_1 x)} + \frac{1}{1+{\lambda_1'}^2} + \frac{1}{1+{\lambda_2'}^2}, \end{aligned}$$

where

$$x = \frac{1}{1+{\lambda_1'}^2} + \frac{{\lambda_2'}^2}{{\lambda_1'}^2(1+{\lambda_2'}^2)}.$$

Exploiting the identity

$$\lambda_1'^{\,2} x + \frac{1}{1+\lambda_1'^{\,2}} + \frac{1}{1+\lambda_2'^{\,2}} = 2,$$

it follows that

$$C' = 2 - \frac{\bar{P}\lambda_1'^{\,2}}{\omega_1} + \frac{\bar{P}^2\lambda_1'^{\,2}}{\omega_1(\bar{P}+\omega_1 x)}.$$

Exchanging the values of $\pi(1)$ and $\pi(2)$ leads to the following permuted version of (6.2):

$$\begin{aligned} &\min_{t_1,t_2} \quad \frac{\omega_1 + t_1}{\omega_1 + (1+\lambda_2'^{\,2})t_1} + \frac{\omega_2 + t_2}{\omega_2 + (1+\lambda_1'^{\,2})t_2} \\ &\text{subject to} \quad t_1 + t_2 = \bar{P}, \quad t_1 \geq 0, \quad t_2 \geq 0. \end{aligned} \tag{6.5}$$

The choice $t_1 = \bar{P}$ and $t_2 = 0$ is feasible in (6.5). Hence, an upper bound C^+ for the optimal objective function value is

$$C^+ = 1 + \frac{\omega_1 + \bar{P}}{(\omega_1 + \bar{P}) + \bar{P}\lambda_2'^{\,2}} = 2 - \frac{\bar{P}\lambda_2'^{\,2}}{\omega_1 + \bar{P}(1+\lambda_2'^{\,2})} = 2 - \frac{\bar{P}\lambda_2'^{\,2}}{\omega_1} + O(\bar{P}^2).$$

Since $\omega_1 < \omega_2$, it follows from the condition $\omega_1\lambda_2'^{\,2} = \omega_2\lambda_1'^{\,2}$ that $\lambda_1'^{\,2} < \lambda_2'^{\,2}$. Comparing C' and C^+, we conclude that for P sufficiently small, $C^+ < C'$, which contradicts the optimality of C'. This completes the proof. □

7. Numerical experiments. Some small test problems were solved to see how P should be chosen in order to observe Theorems 5.1 and 6.1, and to evaluate a conjecture concerning the structure of the optimal permutation in general. In the first experiment, we randomly generate $\omega_i \in [0,1]$ and $\lambda_i \in [0,2]$ in the special case $l = m = n = 10$. The interval $[0,2]$ for $\boldsymbol{\lambda}$ was chosen so that λ_i would be generated on each side of the maximum $x = 1$ for the function $x/(1+x^2)$. These dimensions are small enough that we can enumerate all permutations $\pi \in \mathcal{P}_5$ and select the best. Table 7.1 shows how many times the solution given in Theorems 5.1 or 6.1 is correct for 100 randomly generated problems and for various choices of P.

In another series of experiments, we evaluated the quality of the following M permutations: For each $k = 1, 2, \ldots, M$, let π_k be the permutation defined by

$$\pi_k(i) = i \text{ for } i > k, \quad \pi_k(i) \in [1,k] \text{ for } i \in [1,k], \tag{7.1}$$

TABLE 7.1

Number of times the permutation given by Theorem 5.1 *or* 6.1 *was exact out of* 100 *trials* ($\omega_i \in [0,1]$ *and* $\lambda_i \in [0,2]$, $l = m = n = 10$).

P	Thm. 5.1 exact (out of 100)	Thm. 6.1 exact (out of 100)
10^4	100	0
10^3	97	0
10^2	68	0
10^1	15	0
10^0	1	0
10^{-1}	0	0
10^{-2}	0	44
10^{-3}	0	98
10^{-4}	0	100

TABLE 7.2
Number of times that one of the permutations π_1, π_2, ..., π_M was optimal in (5.1) *out of* 100 *trials* ($\omega_i \in [0,1]$ *and* $\lambda_i \in [0,2]$, $l = m = n = 10$).

P	Some π_k exact (out of 100)	Relative error (no π_k exact)
10^4	100	0
10^3	98	7.8e−08
10^2	74	2.6e−06
10^1	46	5.0e−05
10^0	92	3.0e−05
10^{-1}	96	4.9e−05
10^{-2}	100	0
10^{-3}	100	0
10^{-4}	100	0

and

$$\frac{\lambda_{\pi_k(1)}}{1+\lambda^2_{\pi_k(1)}} \geq \frac{\lambda_{\pi_k(2)}}{1+\lambda^2_{\pi_k(2)}} \geq \ldots \geq \frac{\lambda_{\pi_k(k)}}{1+\lambda^2_{\pi_k(k)}}. \tag{7.2}$$

We optimized (5.1) with the added constraint that π was one of the M permutation π_k, $k = 1, 2, \ldots, M$. In Table 7.2 we consider the same set of test problems used for Table 7.1, and we evaluate the number of times that one of these M permutation yields the exact minimizer. When none of these M permutations yields the exact minimum, we evaluate the relative error in the cost (best approximate cost minus exact cost divided by the exact cost). The average relative error for the best inexact approximation in the set π_k, $1 \leq k \leq M$, is shown in the last column of Table 7.2. Thus, one of the π_k often yields the optimal solution of (5.1). When none of the π_k approximations is optimal, the best approximate cost is nearly optimal.

The motivation for considering the permutations π_k is the following: By Theorem 2.6, there exists an integer $p \geq 1$ (p is the number of positive components of $\boldsymbol{\sigma}$ in an optimal solution) with the property that $\pi(i) = i$ when π is optimal in (5.1) and $i > p$. Hence, we try M different permutations of the form (7.1). When the power P is sufficiently large, we know that the permutation (5.2) is optimal. Thus, we try the same ordering, but applied to the k largest singular values as in (7.2).

8. Conclusions. We analyze the optimization problem (1.4) which arises in linear Bayesian estimation in the presence of noise, and which is relevant to multisensor data fusion problems and wireless communication. Unlike our earlier work [2, 5], we now take into account the noise term $\boldsymbol{\eta}$ in the model (1.1).

By letting the covariance of $\boldsymbol{\eta}$ tends to zero, the results given in the present paper include the results given in [2, 5]. In particular, if we take $\mathbf{N} = \alpha\mathbf{I}$ in (1.3), then a rescaling of $\mathbf{P}$ and $\mathbf{H}$ yields (1.4). In the rescaled problem, the singular values of $\mathbf{H}$ are divided by $\sqrt{\alpha}$. Hence, as α tends to zero, all the singular values in the rescaled problem become larger than 1. Since the function $x/(1+x^2)$ is monotone decreasing for $x > 1$, we deduce that for P sufficiently large, the optimal permutation arranges the singular values in increasing order, the same ordering derived in [2] when the noise $\boldsymbol{\eta}$ was neglected.

For general P, computing the optimal permutation π in (5.1) may not be easy. Nonetheless, we exhibit in section 7 a set of M permutations $\pi_1, \pi_2, \ldots, \pi_M$ which often contains the optimal permutation.

In the case $\mathbf{H} = \mathbf{I}$, the analysis in this paper also yields the results in [5] by taking $\mathbf{N} = \alpha\mathbf{I}$ and letting α tend to zero. It is interesting to note that the analysis in [5] for the case $\boldsymbol{\eta} = \mathbf{0}$ was much more difficult than the analysis of the case $\boldsymbol{\eta} \neq \mathbf{0}$ considered in this paper. Hence, by including the noise term $\boldsymbol{\eta}$ in the model and by letting $\boldsymbol{\eta}$ tend to $\mathbf{0}$, we could recover with less effort the solution given in [5].

9. Appendix. First-order optimality condition. We evaluate the derivative of the Lagrangian (2.5) and set it to zero. Since tr $(\mathbf{A} + \mathbf{A}^*) = 2(\text{Real } [\text{tr } (\mathbf{A})])$ and tr $(\mathbf{AB}) = \text{tr } (\mathbf{BA})$, it follows that the derivative of $\mathbf{SS}^*$ in the direction $\boldsymbol{\delta}\mathbf{S}$ is

$$\text{tr } (\mathbf{S}\boldsymbol{\delta}\mathbf{S}^* + \boldsymbol{\delta}\mathbf{S}\mathbf{S}^*) = 2(\text{Real } [\text{tr } \boldsymbol{\delta}\mathbf{S}\mathbf{S}^*]) = 2(\text{Real } [\text{tr } \mathbf{S}^*\boldsymbol{\delta}\mathbf{S}]). \tag{9.1}$$

For any invertible matrix $\mathbf{M}$, we have

$$\frac{d\mathbf{M}^{-1}}{d\mathbf{T}} = -\mathbf{M}^{-1}\left(\frac{d\mathbf{M}}{d\mathbf{T}}\right)\mathbf{M}^{-1}. \tag{9.2}$$

We equate to zero the derivative of the Lagrangian in the direction $\boldsymbol{\delta}\mathbf{S}$ and utilize (9.1) and (9.2) to obtain

$$\text{Real } \left[\text{tr } \left((\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})\boldsymbol{\Lambda}\mathbf{M}^{-2}\boldsymbol{\Lambda}\mathbf{S}^*\mathbf{L}^{-1} - \mu\mathbf{S}^*\right)\boldsymbol{\delta}\mathbf{S}\right] = \mathbf{0},$$

where $\mathbf{L}$ and $\mathbf{M}$ are defined in (2.4). Inserting

$$\boldsymbol{\delta}\mathbf{S} = \left((\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})\boldsymbol{\Lambda}\mathbf{M}^{-2}\boldsymbol{\Lambda}\mathbf{S}^*\mathbf{L}^{-1} - \mu\mathbf{S}^*\right)^*$$

gives

$$(\mathbf{I} - \mathbf{S}^*\mathbf{L}^{-1}\mathbf{S})\boldsymbol{\Lambda}\mathbf{M}^{-2}\boldsymbol{\Lambda}\mathbf{S}^*\mathbf{L}^{-1} = \mu\mathbf{S}^*.$$

Acknowledgments. We thank the reviewers for their comments and suggestions which led to a more precise and complete paper.

REFERENCES

[1] T. M. Cover and J. A. Thomas, *Elements of Informaton Theory*, John Wiley, New York, 1991.

[2] W. W. Hager, Y. Liu, and T. F. Wong, *Optimization of generalized mean square error in signal processing and communication*, Linear Algebra Appl., 416 (2006), pp. 815–834.

[3] W. W. Hager, *Updating the inverse of a matrix*, SIAM Rev., 31 (1989), pp. 221–239.

[4] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, Philadelphia, 1993.

[5] Y. Liu, T. F. Wong, and W. W. Hager, *Training signal design for estimation of correlated MIMO channels with colored interference*, IEEE Trans. Signal Process., 55 (2007), pp. 1486–1497.

[6] Z.-Q. Luo, G. B. Giannakis, and S. Zhang, *Optimal linear decentralized estimation in a bandwith constrained sensor network*, in Proceedings of the International Conference on Information Theory, Adelaide, Australia, IEEE, 2005.

[7] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, New York, 1979.

[8] D. Palomar, J. Cioffi, and M. Lagunas, *Joint Tx-Rx beamforming design for multicarrier MIMO channels: A unified framework for convex optimization*, IEEE Trans. Signal Process., 51 (2003), pp. 2381–2401.

[9] A. Scaglione, P. Stoica, S. Barbarossa, G. B. Giannakis, and H. Sampath, *Optimal designs for space-time linear precoders and decoders*, IEEE Trans. Signal Process., 50 (2002), pp. 1051–1064.

[10] E. Song, Y. Zhu, and J. Zhou, *Sensors' optimal dimensionality compression matrix in estimation fusion*, Automatica, 41 (2005), pp. 2131–2139.
[11] G. Strang, *Linear Algebra and Its Applications*, 4th ed., Thomson, Belmont, CA, 2006.
[12] P. Viswanath and V. Anantharam, *Optimal sequences for CDMA under colored noise: A Schur-saddle function property*, IEEE Trans. Inform. Theory, 48 (2002), pp. 1295–1318.
[13] A. J. Viterbi, *CDMA-Principles of Spread Spectrum Communication*, Addison-Wesley, Reading, MA, 1995.
[14] T. F. Wong and B. Park, *Training sequence optimization in MIMO systems with colored interference*, IEEE Trans. Commun., 52 (2004), pp. 1939–1947.
[15] Y. Zhu, E. Song, J. Zhou, and Z. You, *Optimal dimensionality reduction of sensor data in multisensor estimation fusion*, IEEE Trans. Signal Process., 53 (2005), pp. 1631–1639.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 805–821

THE EFFECT OF AGGRESSIVE EARLY DEFLATION ON THE CONVERGENCE OF THE QR ALGORITHM*

DANIEL KRESSNER†

Abstract. Aggressive early deflation has proven to significantly enhance the convergence of the QR algorithm for computing the eigenvalues of a nonsymmetric matrix. One purpose of this paper is to point out that this deflation strategy is equivalent to extracting converged Ritz vectors from certain Krylov subspaces. As a special case, the single-shift QR algorithm enhanced with aggressive early deflation corresponds to a Krylov subspace method whose starting vector undergoes a Rayleigh-quotient iteration. It is shown how these observations can be used to derive improved convergence bounds for the QR algorithm.

Key words. QR algorithm, deflation, Krylov subspace, convergence

AMS subject classifications. 65F15, 41A10

DOI. 10.1137/06067609X

1. Introduction.

Let A be a complex $n \times n$ matrix. The aim of the QR algorithm is to compute a Schur decomposition $S = Q^{\mathsf{H}}AQ$, where $Q \in \mathbb{C}^{n\times n}$ is unitary and $S \in \mathbb{C}^{n\times n}$ is upper triangular. The QR algorithm, as introduced by Francis [13, 14] and Kublanovskaya [20], is an iterative process that generates a sequence of unitarily similar matrices $A_0 \leftarrow A, A_1, A_2, \ldots$. Before each iteration, m so-called shifts $\sigma_1, \ldots, \sigma_m \in \mathbb{C}$ are skillfully chosen, defining the shift polynomial $p_i(\lambda) = (\lambda - \sigma_1)\cdots(\lambda - \sigma_m)$. The QR decomposition of $p_i(A_{i-1})$ determines the unitary similarity transformation that yields the next iterate:

$$\begin{aligned} p_i(A_{i-1}) &= Q_i R_i, \qquad \text{(QR decomposition)}, \\ A_i &\leftarrow Q_i^{\mathsf{H}} A_{i-1} Q_i. \end{aligned} \tag{1}$$

Any practically viable implementation of (1) contains at least two further ingredients: initial reduction to condensed form and deflation.

In the following, we assume that A is already in upper Hessenberg form [15]. It is well known that this condensed form is preserved during the iteration (1) and helps greatly to reduce its computational cost. To be more precise, the implicit Q theorem [15] implies that if the Hessenberg matrix A_{i-1} is unreduced (all subdiagonal entries are different from zero), then (1) is equivalent to reducing $V^{\mathsf{H}}A_{i-1}V$ back to Hessenberg form, where V is a unitary matrix that maps the first column of $p_i(A_{i-1})$ to a scalar multiple of the first unit vector e_1. Such an implicit shifted QR iteration requires only $O(mn^2)$ flops (floating point operations), compared to $O(mn^3)$ flops needed by a literal implementation of (1).

As the QR algorithm proceeds, one or more subdiagonal entries of A_i are expected to approach zero. For example, if $\sigma_1, \ldots, \sigma_m$ are chosen to be the eigenvalues of the trailing $m \times m$ principal submatrix of the current iterate, then—under some mild extra assumptions—the $(n-m+1, n-m)$ subdiagonal entry of A_i converges quadratically

*Received by the editors November 28, 2006; accepted for publication (in revised form) by M. Embree May 30, 2008; published electronically September 4, 2008. This work was supported by a DFG Emmy Noether fellowship and in part by the Swedish Foundation for Strategic Research under the Frame Programme grant A3 02:128.
http://www.siam.org/journals/simax/30-2/67609.html
†Seminar für angewandte Mathematik, ETH Zürich, Switzerland (kressner@math.ethz.ch).

to zero [31]. The classical deflation criterion is to consider a subdiagonal element $a^{(i)}_{l,l+1}$ negligible if it satisfies

$$|a^{(i)}_{l,l+1}| \leq \mathbf{u}\big(|a^{(i)}_{l,l}| + |a^{(i)}_{l+1,l+1}|\big), \tag{2}$$

where $a^{(i)}_{kl}$ denotes the (k,l) entry of A_i and $\mathbf{u}$ the unit roundoff. A negligible subdiagonal entry is set to zero, effectively bringing A_i to block upper triangular form:

$$A_i = \begin{bmatrix} A^{(i)}_{11} & A^{(i)}_{12} \\ 0 & A^{(i)}_{22} \end{bmatrix}, \quad A^{(i)}_{11} \in \mathbb{C}^{l\times l}, \quad A^{(i)}_{22} \in \mathbb{C}^{(n-l)\times(n-l)}.$$

This allows one to apply all subsequent QR iterations to the diagonal blocks $A^{(i)}_{11}$ and $A^{(i)}_{22}$ separately and therefore deflate the problem of computing the Schur decomposition of an $n \times n$ matrix into two smaller problems. The QR algorithm is said to have converged when all deflated diagonal blocks are 1×1.

The state-of-the-art LAPACK implementation of the QR algorithm attains high performance by making use of level 3 BLAS operations [6, 21] and employing additional deflation criteria going far beyond the classical criterion (2). Specifically, the [illegible] strategy developed by Braman, Byers, and Mathias [7] often detects converged eigenvalues much earlier than (2) and therefore significantly decreases the overall number of QR iterations needed until convergence. In this paper, we approach this deflation technique from a rather different direction, based on Krylov subspace relations implicitly maintained during the QR algorithm. It turns out that aggressive early deflation amounts to finding and extracting converged Ritz pairs from a Krylov subspace $\mathcal{K}_w(A^{\mathsf{H}}, u_n)$, where u_n denotes the last column of the accumulated unitary transformation matrix. This not only complements the analyses in [7, 33], partially explaining the remarkable success of aggressive early deflation, but also allows for improved convergence bounds. In particular, we can combine the classical convergence theory of the QR algorithm [31] with the convergence of Krylov subspaces to an invariant subspace [4, 5, 25]. The obtained convergence bounds clearly exhibit the benefits of aggressive early deflation.

The rest of this paper is organized as follows. In section 2, we explore different Krylov subspaces associated with the QR algorithm. The Krylov–Schur algorithm, a reliable means to extract and lock converged Ritz pairs from these Krylov subspaces, is recalled in section 3. Reinterpreting this algorithm in terms of unitary transformations on the QR iterate A_i in section 4 reveals its equivalence to aggressive early deflation. Finally, in section 5 we use this relationship to derive convergence bounds for the QR algorithm with aggressive early deflation. In the following, $\|\cdot\|$ denotes the 2-norm of a vector or matrix while $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.

2. Krylov subspace relations. The QR algorithm can be viewed as a nested subspace iteration [3, 8, 24, 29, 32]. Well suited for theoretical purposes, this approach forms the basis of the elegant convergence theory developed by Watkins and Elsner [31]. Starting with a linear subspace $\mathcal{S}_0 \subseteq \mathbb{C}^n$, it can be shown that the QR iteration (1) effects a subspace iteration of the form

$$\mathcal{S}_i = p_i(A)\mathcal{S}_{i-1}, \qquad i = 1, 2, \ldots. \tag{3}$$

If we let $\hat{p}_i = p_i p_{i-1} \cdots p_1$ denote the product of the shift polynomials, then

$$\mathcal{S}_i = \hat{p}_i(A)\mathcal{S}_0, \qquad i = 1, 2, \ldots. \tag{4}$$

Setting $A_0 \equiv A$, we can define

$$\mathcal{S}_0 = \operatorname{span}\{e_1, e_2, \ldots, e_k\},$$

where k satisfies $1 \leq k \leq n$ and e_j denotes the jth unit vector of appropriate length. It is important to note that (3) and (4) hold for all k [illegible] (giving rise to a [illegible] subspace iteration).

To avoid technical difficulties, we assume for the rest of this section that each $p_i(A)$ is nonsingular (a singular $p_i(A)$ results in sudden convergence), which is equivalent to requiring that none of the zeros of p_i coincides with an eigenvalue of A. Then the concrete relation of (3) to the QR iteration (1) is revealed by

$$\mathcal{S}_i = \operatorname{span}\{\hat{Q}_i e_1, \hat{Q}_i e_2, \ldots, \hat{Q}_i e_k\} \tag{5}$$

for $i \geq 0$ with $\hat{Q}_i = Q_1 \cdots Q_i$. Here, $Q_1, \ldots, Q_i$ are the unitary matrices computed during the QR iteration while $\hat{Q}_0$ is defined to be the identity I_n. A simple way to show (5) is to note that (1) implies a QR decomposition

$$\hat{p}_i(A) = \hat{Q}_i (R_i R_{i-1} \cdots R_1) \tag{6}$$

with nonsingular, upper triangular $R_i R_{i-1} \cdots R_1$.

That the implicit shifted QR algorithm operates on Hessenberg matrices links it intimately to Krylov subspace methods; see, e.g., [30] for a recent discussion. Additionally, assuming that A is in unreduced Hessenberg form, it is well known that

$$\mathcal{S}_i = \mathcal{K}_k(A, u_1) = \operatorname{span}\{u_1, Au_1, \ldots, A^{k-1}u_1\}, \tag{7}$$

where u_1 denotes the first column of $\hat{Q}_i$. In fact, if we let u_j denote the jth column of $\hat{Q}_i$ and partition

$$A_i = \hat{Q}_i^{\mathsf{H}} A \hat{Q}_i = \begin{array}{c} \\ k \\ 1 \\ n-k-1 \end{array} \begin{array}{c} \begin{array}{ccc} k & 1 & n-k-1 \end{array} \\ \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{bmatrix} \end{array}, \tag{8}$$

then trivially

$$A[u_1, u_2, \ldots, u_k] = [u_1, u_2, \ldots, u_k] H_{11} + u_{k+1} H_{21}. \tag{9}$$

Under the given assumptions, A_i is in unreduced Hessenberg form and hence the relation (9) happens to be an unreduced Arnoldi decomposition, which implies (7) by [26, Theorem 5.1.1].

Although occasionally mentioned in the literature [10, 29], it is less well known that $\mathcal{S}_i^{\perp}$, the orthogonal complement of $\mathcal{S}_i$, is also a Krylov subspace. To show this, we employ the "flip transpose" of a matrix. Given an $l \times m$ matrix B, let $B^{\mathsf{F}} = F_m B^{\mathsf{H}} F_l$, where F_j denotes the $j \times j$ flip matrix, having ones on the antidiagonal and zeros everywhere else. It can be directly seen that a square matrix B^{F} is in upper Hessenberg (triangular) form if and only if B is in upper Hessenberg (triangular) form. Setting $w = n - k - 1$, the partitioning (8) immediately implies

$$A^{\mathsf{H}}[u_{n-w+1}, \ldots, u_{n-1}, u_n] = [u_{n-w+1}, \ldots, u_{n-1}, u_n] H_{33}^{\mathsf{H}} + u_{n-w} H_{32}^{\mathsf{H}}$$

and, after applying the flip matrix $F \equiv F_w$,

$$A^{\mathsf{H}}[u_n, u_{n-1}, \ldots, u_{n-w+1}] = [u_n, u_{n-1}, \ldots, u_{n-w+1}]H_{33}^{\mathsf{F}} + u_{n-w}H_{32}^{\mathsf{H}}F. \tag{10}$$

Again, (10) is an unreduced Arnoldi decomposition, and therefore

$$\operatorname{span}\{u_n, u_{n-1}, \ldots, u_{n-w+1}\} = \mathcal{K}_w(A^{\mathsf{H}}, u_n)$$

holds for every $w = 1, \ldots, n$. Adjusting w to $w = n - k$, this proves that $\mathcal{S}_i^{\perp}$ indeed coincides with the Krylov subspace $\mathcal{K}_{n-k}(A^{\mathsf{H}}, u_n)$. Since $\hat{p}_i(A)$ is invertible, (6) shows that u_n is parallel to $(\hat{p}_i(A)^{-1})^{\mathsf{H}}e_n =: \hat{p}_i(A)^{-\mathsf{H}}e_n$. Therefore, using the fact that A and $\hat{p}_i(A)^{-1}$ commute,

$$\mathcal{K}_w(A^{\mathsf{H}}, u_n) = \mathcal{K}_w(A^{\mathsf{H}}, \hat{p}_i(A)^{-\mathsf{H}}e_n) = \hat{p}_i(A)^{-\mathsf{H}}\,\mathcal{K}_w(A^{\mathsf{H}}, e_n).$$

3. Extracting Ritz pairs from Krylov subspaces. Given an Arnoldi decomposition of the form (10), a conventional way to extract approximations to eigenvectors from the corresponding Krylov subspace is to compute Ritz pairs and check their residuals. A [illegible] λ from the subspace $\mathcal{K}_w(A^{\mathsf{H}}, u_n)$ is defined as an eigenvalue of the $w \times w$ matrix H_{33}^{F}. The corresponding [illegible] is given by $x = U_w z$, where z is an eigenvector of H_{33}^{F} belonging to λ and $U_w = [u_{n-w+1}, \ldots, u_{n-1}, u_n]$. Taken together, (λ, x) form a so-called [illegible].

With the normalization $\|x\| = \|z\| = 1$, a Ritz pair is usually regarded as converged toward an eigenpair of A^{H} if the norm of the residual $r = A^{\mathsf{H}}x - \lambda x$ is sufficiently small. A practical criterion for deciding upon smallness can be found in the ARPACK manual [23, section 4.6]:

$$\|r\| \leq \max\{\mathbf{u}\|H_{33}^{\mathsf{F}}\|_F, \mathrm{tol} \times |\lambda|\}, \tag{11}$$

where $\mathbf{u}$ denotes the unit roundoff and tol is a tolerance chosen by the user. Note that (10) implies $r = (H_{32}^{\mathsf{H}}Fz)u_{n-w}$ and hence $\|r\| = |H_{32}^{\mathsf{H}}Fz|$. For tol $= 0$, the criterion (11) yields normwise backward stability: (λ, x) is the exact eigenpair of the perturbed matrix $A^{\mathsf{H}} + (\triangle A)^{\mathsf{H}}$, where

$$\triangle A = -xr^{\mathsf{H}} = -(z^{\mathsf{H}}FH_{32})(U_w z)u_{n-w}^{\mathsf{H}} \tag{12}$$

satisfies $\|\triangle A\|_F = \|r\| \leq \mathbf{u}\|H_{33}^{\mathsf{F}}\|_F \leq \mathbf{u}\|A\|_F$.

3.1. Locking converged Ritz values. Stewart's [illegible] [27] provides a numerically reliable means to detect, extract, and lock converged Ritz pairs. For some Ritz value λ, an ordered Schur decomposition [15] of H_{33}^{F} is computed such that λ appears in the top left corner of the triangular factor:

$$V^{\mathsf{H}}H_{33}^{\mathsf{F}}V = \begin{bmatrix} \lambda & T_{12} \\ 0 & T_{22} \end{bmatrix}, \tag{13}$$

where $V \in \mathbb{C}^{w \times w}$ is unitary and $T_{22} \in \mathbb{C}^{(w-1)\times(w-1)}$. A corresponding Ritz vector is given by $x = U_w z$ with $z = Ve_1$. If (11) is satisfied, we partition $U_w V = [x, \widehat{U}_{w-1}]$ and

$$H_{32}^{\mathsf{H}}FV = [\bar{s}_1, s_2^{\mathsf{H}}], \tag{14}$$

where $s_1 = z^{\mathsf{H}} F H_{32}$. Together with (10), this implies

$$A^{\mathsf{H}}[x, \widehat{U}_{w-1}] = [x, \widehat{U}_{w-1}] \begin{bmatrix} \lambda & T_{12} \\ 0 & T_{22} \end{bmatrix} + u_{n-w}[\bar{s}_1, s_2^{\mathsf{H}}]. \tag{15}$$

If (11) is not satisfied, we can test any other Ritz pair by reordering the eigenvalues in the Schur form (13).

Continuing the described process for the Ritz values contained in T_{22} eventually leads to a decomposition of the form

$$A^{\mathsf{H}}[X, \widehat{U}_{w-d}] = [X, \widehat{U}_{w-d}] \begin{bmatrix} \widehat{T}_{11} & \widehat{T}_{12} \\ 0 & \widehat{T}_{22} \end{bmatrix} + u_{n-w}[\hat{s}_1^{\mathsf{H}}, \hat{s}_2^{\mathsf{H}}], \tag{16}$$

where the upper triangular matrix $\widehat{T}_{11}$ contains all d converged Ritz values on its diagonal and

$$\|\hat{s}_1\| \le \sqrt{d} \max\{\mathbf{u}, \mathsf{tol}\} \|H_{33}^{\mathsf{F}}\|_F. \tag{17}$$

The matrices U_w and $[X, \widehat{U}_{w-d}]$ span the same space, i.e., there is a unitary matrix $\widehat{V}$ such that $[X, \widehat{U}_{w-d}] = U_w \widehat{V}$.

3.2. Restoring the Arnoldi decomposition. Note that (16) is not a standard Arnoldi decomposition, which would require $[\hat{s}_1^{\mathsf{H}}, \hat{s}_2^{\mathsf{H}}]$ to be a multiple of e_w^{H}. The vector $\hat{s}_1^{\mathsf{H}}$ is negligible but $\hat{s}_2^{\mathsf{H}}$ is not. Therefore, the last (optional) step of the Krylov–Schur algorithm consists of transforming $\hat{s}_2^{\mathsf{H}}$ back to a multiple of e_{w-d}^{H} while maintaining $\left[\begin{smallmatrix} \widehat{T}_{11} & \widehat{T}_{12} \\ 0 & \widehat{T}_{22} \end{smallmatrix}\right]$ in upper Hessenberg form.

To achieve this goal, first a unitary matrix V_1 is computed such that $V_1^{\mathsf{H}} \hat{s}_2 = \beta e_{w-d}$ with $|\beta| = \|\hat{s}_2\|$. By a row-oriented version of the usual Hessenberg reduction algorithm [26, p. 312], a unitary matrix $V_2 = \tilde{V}_2 \oplus 1$ can be computed such that $\widetilde{T}_{22} = V_2^{\mathsf{H}}(V_1^{\mathsf{H}} \widehat{T}_{22} V_1) V_2$ is again in Hessenberg form. Setting $\widetilde{T}_{12} = \widehat{T}_{12} V_1 V_2$ and $\widetilde{U}_{w-d} = \widehat{U}_{w-d} V_1 V_2$ finally yields

$$A^{\mathsf{H}}[X, \widetilde{U}_{w-d}] = [X, \widetilde{U}_{w-d}] \begin{bmatrix} \widehat{T}_{11} & \widetilde{T}_{12} \\ 0 & \widetilde{T}_{22} \end{bmatrix} + u_{n-w}[\hat{s}_1^{\mathsf{H}}, \bar{\beta} e_{w-d}^{\mathsf{H}}], \tag{18}$$

which becomes an Arnoldi decomposition after setting $\hat{s}_1$ to zero.

4. Aggressive early deflation. In this section, we reinterpret the extraction of Ritz pairs from the Arnoldi decomposition (10) in terms of transformations operating on the $n \times n$ Hessenberg matrix A_i. Let us recall the partitioning (8) with $w = n-k-1$:

$$A_i = \hat{Q}_i^{\mathsf{H}} A \hat{Q}_i = \begin{array}{c} \\ w \\ 1 \\ n-w-1 \end{array} \begin{array}{c} \begin{array}{ccc} n-w-1 & 1 & w \end{array} \\ \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{bmatrix} \end{array}.$$

In the context of the QR algorithm, we refer to w as the [illegible] and assume $w \ll n$ to make sure that the cost of the extraction process remains modest in comparison to the cost of a QR iteration.

4.1. Locking converged Ritz values. Note that all variables used in this section refer precisely to the same quantities introduced in section 3. A Ritz value λ has been defined to be an eigenvalue of H_{33}^{F}, and z denotes a corresponding normalized eigenvector. It directly follows that $\bar{\lambda}$ is an eigenvalue of H_{33} having $F_w z$ as a corresponding eigenvector. Setting $V_{\mathsf{F}} = F_w V F_w$, the ordered Schur decomposition (13) implies

$$V_{\mathsf{F}}^{\mathsf{H}} H_{33} V_{\mathsf{F}} = F_w \left(V^{\mathsf{H}} H_{33}^{\mathsf{F}} V\right)^{\mathsf{H}} F_w = F_w \begin{bmatrix} \bar{\lambda} & 0 \\ T_{12}^{\mathsf{H}} & T_{22}^{\mathsf{H}} \end{bmatrix} F_w = \begin{bmatrix} T_{22}^{\mathsf{F}} & F_{w-1} T_{12}^{\mathsf{H}} \\ 0 & \bar{\lambda} \end{bmatrix}, \tag{19}$$

which is an ordered Schur decomposition for H_{33}. Moreover, it follows from (14) that

$$V_{\mathsf{F}}^{\mathsf{H}} H_{32} = F_w V^{\mathsf{H}} (F_w H_{32}) = F_w \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} F_{w-1} s_2 \\ s_1 \end{bmatrix}.$$

These two relations yield

$$(I \oplus V_{\mathsf{F}})^{\mathsf{H}} A_i (I \oplus V_{\mathsf{F}}) = \begin{bmatrix} H_{11} & H_{12} & \check{H}_{13} & \check{H}_{14} \\ H_{21} & H_{22} & \check{H}_{23} & \check{H}_{24} \\ 0 & F_{w-1} s_2 & T_{22}^{\mathsf{F}} & T_{12}^{\mathsf{F}} \\ 0 & s_1 & 0 & \bar{\lambda} \end{bmatrix},$$

where we define $\left[\begin{smallmatrix} \check{H}_{13} & \check{H}_{14} \\ \check{H}_{23} & \check{H}_{24} \end{smallmatrix}\right] := \left[\begin{smallmatrix} H_{13} \\ H_{23} \end{smallmatrix}\right] V_{\mathsf{F}}$. Note that the vector $\left[\begin{smallmatrix} F_{w-1} s_2 \\ s_1 \end{smallmatrix}\right]$ is called the [illegible] in [7].

Let us recall that the residual r of the Ritz value λ satisfies $r = (H_{32}^{\mathsf{H}} F z) u_{n-w} = \bar{s}_1 u_{n-w}$, and hence $\|r\| = |s_1|$. Thus if (11) is satisfied for $\mathrm{tol} = 0$, the trailing spike element can be safely set to zero and $\bar{\lambda}$ can be regarded as a computed eigenvalue of A without spoiling the numerical backward stability of the QR algorithm. If $|s_1| > \mathbf{u}\|H_{33}\|_F$, we can test any other eigenvalue of H_{33} by considering a differently ordered Schur decomposition. This is equivalent to the search for converged Ritz values, since the ordered Schur decompositions of H_{33} and H_{33}^{F} are connected to each other in the one-to-one relationship (19).

Extracting and locking further converged Ritz values of H_{33}^{F}, as described in section 3, eventually yields a unitary matrix $\widehat{V}_{\mathsf{F}} \in \mathbb{C}^{w \times w}$ such that

$$\widehat{A}_i = (I \oplus \widehat{V}_{\mathsf{F}})^{\mathsf{H}} A_i (I \oplus \widehat{V}_{\mathsf{F}}) = \begin{bmatrix} H_{11} & H_{12} & \widehat{H}_{13} & \widehat{H}_{14} \\ H_{21} & H_{22} & \widehat{H}_{23} & \widehat{H}_{24} \\ 0 & F_{w-d} \hat{s}_2 & \widehat{T}_{22}^{\mathsf{F}} & \widehat{T}_{12}^{\mathsf{F}} \\ 0 & F_d \hat{s}_1 & 0 & \widehat{T}_{11}^{\mathsf{F}} \end{bmatrix}, \tag{20}$$

where the diagonal of the upper triangular matrix $\widehat{T}_{11}^{\mathsf{F}}$ contains the complex conjugates of all converged Ritz values. The d trailing spike elements in $F_d \hat{s}_1$ are all negligible, since (17) amounts to $\|F_d \hat{s}_1\| \le \sqrt{d}\mathbf{u}\|H_{33}\|_F \le \sqrt{d}\mathbf{u}\|A\|_F$ for $\mathrm{tol} = 0$.

4.2. Restoring the Hessenberg form. To continue the QR algorithm, the matrix $\widehat{A}_i$ in (20) needs to be restored to Hessenberg form. This is exactly what is achieved by the unitary transformations defined in section 3.2. Setting $V_{1,\mathsf{F}} = F_{w-d} V_1 F_{w-d}$ and $V_{2,\mathsf{F}} = F_{w-d} V_2 F_{w-d}$ yields

$$V_{1,\mathsf{F}}^{\mathsf{H}} (F_{w-d} \hat{s}_2) = \beta e_1, \quad V_{2,\mathsf{F}}^{\mathsf{H}} (V_{1,\mathsf{F}}^{\mathsf{H}} \widehat{T}_{22}^{\mathsf{F}} V_{1,\mathsf{F}}) V_{2,\mathsf{F}} = \widetilde{T}_{22}^{\mathsf{F}},$$

where $\widetilde{T}_{22}^{\mathsf{F}}$ is in upper Hessenberg form. Hence,

$$\widetilde{A}_i = (I \oplus V_{1,\mathsf{F}} V_{2,\mathsf{F}} \oplus I_d)^{\mathsf{H}} \widehat{A}_i (I \oplus V_{1,\mathsf{F}} V_{2,\mathsf{F}} \oplus I_d) = \begin{bmatrix} H_{11} & H_{12} & \widetilde{H}_{13} & \widehat{H}_{14} \\ H_{21} & H_{22} & \widetilde{H}_{23} & \widehat{H}_{24} \\ 0 & \beta e_1 & \widetilde{T}_{22}^{\mathsf{F}} & \widetilde{T}_{12}^{\mathsf{F}} \\ 0 & F_d \hat{s}_1 & 0 & \widehat{T}_{11}^{\mathsf{F}} \end{bmatrix}$$

with $\left[\begin{smallmatrix} \widetilde{H}_{13} \\ \widetilde{H}_{23} \end{smallmatrix}\right] = \left[\begin{smallmatrix} \widetilde{H}_{13} \\ \widetilde{H}_{23} \end{smallmatrix}\right] V_{1,\mathsf{F}} V_{2,\mathsf{F}}$. Setting $\hat{s}_1$ to zero returns $\widetilde{A}_i$ to Hessenberg form, and the QR algorithm can be continued on the leading $(n-d) \times (n-d)$ principal submatrix of $\widetilde{A}_i$.

4.3. Comparison to classical deflation. The reader is invited to check that the presented deflation algorithm obtained via the extraction of Ritz pairs from the Krylov subspace $\mathcal{K}_w(A^{\mathsf{H}}, u_n)$ is precisely the same as the aggressive early deflation algorithm originally described in [7]. Let us contemplate what the different deflation criteria for the QR algorithm mean in terms of Krylov subspaces.

The classical deflation criterion (2) tests the smallness of $|a_{l,l+1}^{(i)}|$ for each $l = 1, \ldots, n-1$. From the Arnoldi decomposition (10) it is immediately clear that $|a_{l,l+1}^{(i)}|$ is the minimal norm of a backward error matrix $\triangle A$ such that $\mathcal{K}_{n-l}(A^{\mathsf{H}}, u_n)$ becomes an invariant subspace of $(A + \triangle A)^{\mathsf{H}}$. In other words, classical deflation considers the nested sequence of Krylov subspaces

$$\mathcal{K}_1(A^{\mathsf{H}}, u_n),\ \mathcal{K}_2(A^{\mathsf{H}}, u_n), \ldots, \mathcal{K}_{n-1}(A^{\mathsf{H}}, u_n)$$

and checks whether any of them is a good approximation to an invariant subspace of A^{H} as a whole. If the shifts in the QR algorithm are chosen as the eigenvalues of the $m \times m$ trailing submatrix, a choice sometimes called [illegible], then deflation is most likely to happen for $\mathcal{K}_{n-l}(A^{\mathsf{H}}, u_n)$ with $n - l \leq m$, as the convergence theory [31] states that $\mathcal{K}_m(A^{\mathsf{H}}, u_n)$ converges locally quadratically to an invariant subspace of A^{H}. Roughly speaking (neglecting the effects of ill-conditioning), the approximation quality of $\mathcal{K}_{n-l}(A^{\mathsf{H}}, u_n)$ as a whole is determined by the poorest Ritz vector approximation that can be extracted from $\mathcal{K}_{n-l}(A^{\mathsf{H}}, u_n)$. Hence, slowly converging Ritz vectors hinder the deflation of other, quickly converging Ritz vectors. Aggressive early deflation is fundamentally different and avoids this effect. Only one Krylov subspace $\mathcal{K}_w(A^{\mathsf{H}}, u_n)$ for some fixed value of w is considered. Moreover, not the convergence of $\mathcal{K}_w(A^{\mathsf{H}}, u_n)$ as a whole but the convergence of each individual Ritz vector to an eigenvector of A^{H} is checked. Provided that A has distinct eigenvalues, all Ritz vectors from the Krylov subspace $\mathcal{K}_m(A^{\mathsf{H}}, u_n)$ converge locally quadratically, but some may converge at a significantly faster rate and can be deflated much earlier. This is demonstrated by the following example.

[illegible] 1. We applied the QR algorithm with $m = 4$ Francis shifts to a 250×250 random matrix generated by the following MATLAB commands:

```
randn('state',0);   A = hess(randn(250)+1i*randn(250)).
```

Table 1 compares classical with aggressive early deflation and clearly exhibits the advantages of the latter. For example, consider the Krylov subspace $\mathcal{K}_4(A^{\mathsf{H}}, u_n)$ after $i = 3$ QR iterations. The norms of the residuals for the four corresponding Ritz pairs are 3.8×10^{-2}, 1.3×10^{-3}, 6.0×10^{-4}, and 1.2×10^{-6}. The magnitude of the corresponding subdiagonal element, $|a_{n-4,n-3}^{(3)}| = 4.6 \times 10^{-2}$, is nearly the maximum

TABLE 1
Magnitudes of trailing subdiagonal elements (left set of columns) and minimal residuals for the Ritz pairs extracted from $\mathcal{K}_w(A^{\mathsf{H}}, u_n)$ (right set of columns) after i QR iterations with 4 Francis shifts applied to the matrix from Example 1.

	Classical deflation: $\lvert a_{l+1,l}^{(i)}\rvert$, $l =$				Aggressive early deflation: $\min \lVert r\rVert$, $w =$			
	$n-4$	$n-3$	$n-2$	$n-1$	2	4	6	8
$i=0$	$2.3\times10^{+0}$	$1.7\times10^{+0}$	$1.9\times10^{+0}$	$2.1\times10^{+0}$	$1.3\times10^{+0}$	2.2×10^{-1}	9.1×10^{-2}	7.3×10^{-2}
$i=1$	9.1×10^{-1}	$1.2\times10^{+0}$	$1.8\times10^{+0}$	$2.7\times10^{+0}$	$1.2\times10^{+0}$	6.0×10^{-2}	1.5×10^{-2}	6.4×10^{-3}
$i=2$	4.9×10^{-1}	5.8×10^{-1}	$1.3\times10^{+0}$	6.0×10^{-1}	2.1×10^{-1}	2.0×10^{-3}	8.6×10^{-5}	5.0×10^{-5}
$i=3$	4.6×10^{-2}	9.2×10^{-2}	$1.7\times10^{+0}$	1.8×10^{-2}	8.5×10^{-3}	1.2×10^{-6}	9.6×10^{-9}	7.7×10^{-9}

of these numbers, demonstrating that the convergence of $\mathcal{K}_4(A^{\mathsf{H}}, u_n)$ as a whole is determined by the poorest Ritz pair approximation.

In Table 1, aggressive early deflation already shows dramatic improvements for $w = m$, i.e., when the size of the deflation window coincides with the number of Francis shifts. Choosing $w > m$ enlarges the Krylov subspace and adds a Krylov subspace acceleration to the quadratically converging Ritz vectors. This is explored in more detail in the next section.

2. If each QR iteration is based on a single shift that is defined to be the (n,n) entry of the current iterate, then it is well known that u_n undergoes a Rayleigh-quotient iteration; see, e.g., [26, section 2.1]. To be more specific, let $u_n^{(i-1)}$ denote the last column of the accumulated unitary transformation matrix U_{i-1} satisfying $A_{i-1} = U_{i-1}^{\mathsf{H}} A U_{i-1}$. Then

$$u_n^{(i)} = \frac{\left(A^{\mathsf{H}} - \sigma_1^{(i)}\right)^{-1} u_n^{(i-1)}}{\left\| \left(A^{\mathsf{H}} - \sigma_1^{(i)}\right)^{-1} u_n^{(i-1)} \right\|},$$

provided that $\sigma_1^{(i)} = u_n^{(i-1)\mathsf{H}} A u_n^{(i-1)}$ is not an eigenvalue of A^{H}. Using aggressive early deflation, we deflate converged Ritz pairs from the Krylov subspace $\mathcal{K}_w(A^{\mathsf{H}}, u_n^{(i)})$. This shows that the single-shifted QR iteration equipped with such a deflation strategy is, in fact, a Krylov subspace method where the starting vector undergoes a quadratically convergent iteration.

There are situations for which the classical criterion detects deflations that go undetected if only aggressive early deflation is used. Probably the most practically relevant situation is that linear convergence phenomena occur if the shifts vary slightly in the course of several QR iterations. The typical consequence is that one or more of the leading subdiagonal elements of A_i approach zero. As $w \ll n$, the Krylov subspace $\mathcal{K}_w(A^{\mathsf{H}}, u_n)$ will not benefit from this effect. As a remedy, one could additionally use a variant of aggressive early deflation that considers Ritz pairs from $\mathcal{K}_w(A, u_1)$. However, numerical experiments reported in [18] reveal that these linear convergence phenomena seem to be too rare to justify the extra computational effort.

A quite different situation is pointed out in [7, section 2.5] and it is interesting to see this example in the light of Krylov subspaces. Let

$$A = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & \varepsilon & 2 & 0 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}, \tag{21}$$

where $0 < \varepsilon \ll 1$. The norm of the residual for every Ritz pair from $\mathcal{K}_4(A^H, e_5)$ is approximately given by $\sqrt{\varepsilon}$, even though a perturbation of norm ε [illegible] the deflation window deflates two eigenvalues. Caused by the high nonnormality of $A(2:5, 2:5)$, standard Ritz pair extraction fails to detect the converged left eigenvector contained in $\mathcal{K}_4(A^H, e_5)$. Refined Ritz pairs [16, 17] aim to avoid this effect. For each Ritz value λ, the refined Ritz vector is chosen to minimize the norm of the corresponding residual. Unfortunately, for our example each refined Ritz pair has residual norm of about $\sqrt{\varepsilon/2}$. In [19], a computational procedure based on the distance to uncontrollability was described that optimizes both the refined Ritz vector [illegible] the value. For our example, applying this costly procedure leads to a refined Ritz pair whose residual norm nearly attains ε. So, in principle, refined Ritz pairs can be used to improve aggressive early deflation even further. However, preliminary numerical experiments with several matrices from the matrix market collection [2] indicate that a situation like (21) occurs rarely in practice, at least too seldom to justify the extra computational effort needed for extracting refined Ritz pairs.

5. Convergence bounds. To discuss the impact of aggressive early deflation on the convergence of the QR algorithm, let us return to the notation of section 2. Watkins and Elsner [31, Theorem 6.3] have shown that $\mathcal{S}_i = \mathcal{K}_{n-m}(A, u_1)$ converges locally quadratically to an $(n-m)$-dimensional invariant subspace $\mathcal{X}$ of A, provided that A has distinct eigenvalues and that in each iteration m Francis shifts $\sigma_1, \ldots, \sigma_m$ are chosen. To formalize this statement, let us employ the [illegible] between two subspaces $\mathcal{S}$ and $\mathcal{T}$ (not necessarily of the same dimension) defined as

$$d(\mathcal{S}, \mathcal{T}) := \sup_{\substack{s \in \mathcal{S} \\ \|s\|=1}} \inf_{t \in \mathcal{T}} \|s - t\|.$$

For convenience, we write $d(y, \mathcal{T})$ if $\mathcal{S}$ is spanned by a single vector y. With this notation, the convergence statement reads as follows:

[illegible] $d(\mathcal{S}_i, \mathcal{X}) \to 0$ [illegible] $i \to \infty$ [illegible]

At the time of this writing, the task of finding conditions for global convergence, which seems to hold almost always in practice, remains open and is not the subject of this paper.

Relevant properties of the gap $d(\cdot, \cdot)$ are summarized, e.g., in [4, 31]. We have $d(\mathcal{S}, \mathcal{T}) = 0$ if and only if $\mathcal{S} \subseteq \mathcal{T}$. If $\Pi_{\mathcal{S}}$ and $\Pi_{\mathcal{T}}$ denote orthogonal projections onto $\mathcal{S}$ and $\mathcal{T}$, respectively, then $d(\mathcal{S}, \mathcal{T}) = \|(I - \Pi_{\mathcal{T}})\Pi_{\mathcal{S}}\|$. This readily implies $d(\mathcal{T}^\perp, \mathcal{S}^\perp) = d(\mathcal{S}, \mathcal{T})$. The last property allows us to replace $d(\mathcal{S}_i, \mathcal{X})$ by

$$d(\mathcal{Y}, \mathcal{S}_i^\perp) = d\big(\mathcal{Y}, \mathcal{K}_m(A^H, u_n)\big) = d\big(\mathcal{Y}, \hat{p}_i(A)^{-H}\mathcal{K}(A^H, e_n)\big),$$

with $\mathcal{Y} = \mathcal{X}^\perp$, in the convergence discussion of the QR algorithm. Note that $\mathcal{Y}$ is an invariant subspace of A^H, or, equivalently, a [illegible] invariant subspace of A.

5.1. Convergence of eigenvectors. As discussed in section 4.3, aggressive early deflation has two advantages: (1) approximation by Ritz vectors instead of the whole Krylov subspace, and (2) additional Krylov subspace acceleration. The convergence theory by Watkins and Elsner must be modified in order to accommodate both advantages; the following theorem addresses the first one.

THEOREM 3. [illegible] $A \in \mathbb{C}^{n \times n}$ [illegible]

$$Q^H A Q = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad A_{11} \in \mathbb{C}^{(n-m)\times(n-m)}, \quad A_{22} \in \mathbb{C}^{m \times m},$$

[illegible] $Q \in \mathbb{C}^{n\times n}$ [illegible] $\lambda(A_{11}) \cap \lambda(A_{22}) = \emptyset$ [illegible] y [illegible] $\lambda_2 \in \lambda(A_{22})$ [illegible] $\mathcal{Y}$ [illegible] A [illegible] $\lambda(A_{22})$ [illegible] f [illegible] A [illegible]

$$d\big(y, f(A)^{-\mathsf{H}}\mathcal{V}\big) \le \frac{C_T}{1 - C_T\, d(\mathcal{Y},\mathcal{V})}\, \|f(A_{11})^{-1}\|\, |f(\lambda_2)|\, d(y,\mathcal{V}) \tag{22}$$

[illegible] m [illegible] $\mathcal{V}$ [illegible] $C_T\, d(\mathcal{Y},\mathcal{V}) < 1$ [illegible]

$$C_T = \Big(\|P_{A_{11}}\| + \sqrt{1+\|P_{A_{11}}\|^2}\Big)\Big(\|P_{\lambda_2}\| + \sqrt{1+\|P_{\lambda_2}\|^2}\Big).$$

[illegible] $P_{A_{11}} \in \mathbb{C}^{n\times n}$ [illegible] A [illegible] $\lambda(A_{11})$ [illegible] $P_{\lambda_2} \in \mathbb{C}^{m\times m}$ [illegible] A_{22} [illegible] λ_2 [illegible]. In the following, $\kappa(\cdot)$ denotes the 2-norm condition number of a matrix. By [11], there are invertible lower block triangular matrices T_1, T_2 with

$$\kappa(T_1) = \|P_{A_{11}}\| + \sqrt{1+\|P_{A_{11}}\|^2}, \quad \kappa(T_2) = \|P_{\lambda_2}\| + \sqrt{1+\|P_{\lambda_2}\|^2},$$

such that

$$T_1^{-1}\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22}\end{bmatrix} T_1 = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22}\end{bmatrix}, \quad T_2^{-1} A_{22} T_2 = \begin{bmatrix} \widetilde{A}_{22} & 0 \\ 0 & \lambda_2\end{bmatrix}.$$

The matrix $T := QT_1(I \oplus T_2)$ satisfies $\kappa(T) \le \kappa(T_1)\kappa(T_2) = C_T$ and block diagonalizes A:

$$D := T^{-1}AT = A_{11} \oplus \tilde{A}_{22} \oplus \lambda_2. \tag{23}$$

In the following, we first derive a bound for the accordingly transformed subspaces $\widetilde{\mathcal{V}} = T^{\mathsf{H}}\mathcal{V}$ and $f(D)^{-\mathsf{H}}\widetilde{\mathcal{V}}$, which will then be turned into a bound for the original subspaces.

Since $\tilde{y} := T^{\mathsf{H}}y$ is a left eigenvector of the block diagonal matrix $D = T^{-1}AT$ belonging to the simple eigenvalue λ_2, we must have $\tilde{y} = \beta e_n$ for some $\beta \in \mathbb{C}$. To simplify the description, we assume without loss of generality that $\tilde{y} = e_n$. Analogously, we have

$$\widetilde{\mathcal{Y}} := T^{\mathsf{H}}\mathcal{Y} = \operatorname{span}\begin{bmatrix} 0 \\ I_m\end{bmatrix}. \tag{24}$$

By [31, Lemma 4.1], the condition $C_T\, d(\mathcal{Y},\mathcal{V}) < 1$ implies $d(\widetilde{\mathcal{Y}},\widetilde{\mathcal{V}}) < 1$, Hence, no vector in $\widetilde{\mathcal{V}}$ is perpendicular to $\widetilde{\mathcal{Y}}$. Taking (24) into account, this means that no nonzero vector contained in $\widetilde{\mathcal{V}}$ can have m trailing zero entries. Therefore we can choose a basis of the form $V = \left[\begin{smallmatrix} F \\ I_m\end{smallmatrix}\right]$ for $\widetilde{\mathcal{V}}$. The norm of F determines the distance between $\widetilde{\mathcal{V}}$ and $\widetilde{\mathcal{Y}}$. Specifically, we have

$$\|F\| = d(\widetilde{\mathcal{Y}},\widetilde{\mathcal{V}})/\sqrt{1 - d(\widetilde{\mathcal{Y}},\widetilde{\mathcal{V}})^2}; \tag{25}$$

see, e.g., [28].

To obtain a bound on $d\big(e_n, f(D)^{-\mathsf{H}}\widetilde{\mathcal{V}}\big)$, we select the particularly convenient vector

$$v_1 = Ve_m = \begin{bmatrix} Fe_m \\ e_m \end{bmatrix} \in \widetilde{\mathcal{V}}. \tag{26}$$

We have

$$f(D)^{-\mathsf{H}}v_1 = \begin{bmatrix} f(A_{11})^{-\mathsf{H}}Fe_m \\ 1/\overline{f(\lambda_2)}e_m \end{bmatrix},$$

which implies

$$d\big(e_n, f(D)^{-\mathsf{H}}v_1\big) \le \|f(A_{11})^{-\mathsf{H}}Fe_m\||f(\lambda_2)| \le \|f(A_{11})^{-1}\||f(\lambda_2)|\,\|e_n - v_1\|. \tag{27}$$

Note that v_1 is in general not the vector in $\widetilde{\mathcal{V}}$ that is closest to e_n, so we cannot simply replace $\|e_n - v_1\|$ by $d\big(e_n, \widetilde{\mathcal{V}}\big)$ in (27). In fact, the closest vector is the solution of the minimization problem $d(e_n, \widetilde{\mathcal{V}}) = \inf_{v\in\widetilde{\mathcal{V}}} \|e_n - v\|$ and takes the form $v_0 = V(V^{\mathsf{H}}V)^{-1}V^{\mathsf{H}}e_n = V(V^{\mathsf{H}}V)^{-1}e_m$. Lemma 8 from the appendix reveals the relationship

$$\|e_n - v_1\| \le \frac{\sqrt{1+\|F\|^2}}{\sqrt{1+\|F\|^2}-\|F\|}\|e_n - v_0\| = \frac{\|e_n - v_0\|}{1 - d(\widetilde{\mathcal{Y}}, \widetilde{\mathcal{V}})}, \tag{28}$$

where we used (25) for the latter equality.

Combining (27) with (28) shows

$$d\big(e_n, f(D)^{-\mathsf{H}}\widetilde{\mathcal{V}}\big) \le d\big(e_n, f(D)^{-\mathsf{H}}v_1\big) \le \frac{\|f(A_{11})^{-1}\||f(\lambda_2)|}{1 - d(\widetilde{\mathcal{Y}}, \widetilde{\mathcal{V}})} d(\tilde{y}, \widetilde{\mathcal{V}}).$$

Recall that $y = T^{-\mathsf{H}}e_n$, $\mathcal{V} = T^{-\mathsf{H}}\widetilde{\mathcal{V}}$, and $f(A)^{-\mathsf{H}}\mathcal{V} = T^{-\mathsf{H}}f(D)^{-\mathsf{H}}\widetilde{\mathcal{V}}$. Using Lemma 9 from the appendix, we obtain

$$d\big(y, f(A)^{-\mathsf{H}}\mathcal{V}\big) \le C_T \frac{\|f(A_{11})^{-1}\|\,|f(\lambda_2)|}{1 - d(\widetilde{\mathcal{Y}}, \widetilde{\mathcal{V}})} d(y, \mathcal{V}),$$

using $\kappa(T) \le C_T$. The proof is completed after applying the estimate $d(\widetilde{\mathcal{Y}}, \widetilde{\mathcal{V}}) \le C_T\, d(\mathcal{Y}, \mathcal{V})$. □

4. The condition $C_T\, d(\mathcal{Y}, \mathcal{K}_m(A^{\mathsf{H}}, e_n)) < 1$ in Theorem 3 is an artifact of the proof technique and can be removed. Let $\mathcal{U}$ denote the right invariant subspace of A belonging to $\lambda(A_{11})$. Then $\mathcal{U} \cap \mathcal{V} = \{0\}$ is sufficient to guarantee that the condition $d(\widetilde{\mathcal{Y}}, \widetilde{\mathcal{V}}) < 1$ needed in the proof of Theorem 3 is satisfied. Hence, there is a constant C independent of f such that

$$d\big(y, f(A)^{-\mathsf{H}}\mathcal{V}\big) \le C\,\|f(A_{11})^{-1}\|\,|f(\lambda_2)|\,d(y, \mathcal{V}),$$

even under this rather mild assumption. Note that $\mathcal{U} \cap \mathcal{V} = \{0\}$ is always satisfied if A is in unreduced Hessenberg form and $\mathcal{V} = \mathcal{K}_m(A^{\mathsf{H}}, e_n)$ [31].

Theorem 3 is applied to the convergence analysis of the QR algorithm by setting $f = \hat{p}_i$ and $\mathcal{V} = \mathcal{K}_m(A^{\mathsf{H}}, e_n)$. The classical convergence analysis [31, Lemma 4.4] provides bounds of the form

$$d\big(\mathcal{Y}, \hat{p}_i(A)^{-\mathsf{H}}\mathcal{K}_m(A^{\mathsf{H}}, e_n)\big) \le C\,\|\hat{p}_i(A_{11})^{-1}\|\,\|\hat{p}_i(A_{22})\|\,d(\mathcal{Y}, \mathcal{K}_m(A^{\mathsf{H}}, e_n)) \tag{29}$$

for some constant C. [illegible] shifts (i.e., the roots of p_i) need to converge simultaneously to eigenvalues of A_{22} in order to attain superlinear converge. In contrast, the bound (22) predicts superlinear convergence even if only one shift converges to a simple eigenvalue of A_{22}.

Corollary 5. [illegible] 3, [illegible] $p_1, p_2, \ldots$ [illegible] $\hat{p}_i(A)$ [illegible] $\hat{p}_i = p_1 p_2 \cdots p_i$ [illegible] p_j [illegible] λ_2 [illegible] $j \to \infty$ [illegible] $p_j(A_{11})^{-1}$ [illegible] $\rho > 0$ [illegible] C [illegible]

$$d\big(y, \hat{p}_i(A)^{-\mathsf{H}} \mathcal{K}_m(A^{\mathsf{H}}, e_n)\big) \le C\rho^i,$$

[illegible] $\mathcal{U} \cap \mathcal{K}_m(A^{\mathsf{H}}, e_n) = \{0\}$ [illegible] $\mathcal{U}$ [illegible] A [illegible] $\lambda(A_{11})$

[illegible]. Let d be the maximal degree of p_j, and let $C_2 = \max\{1, \widetilde{C}_2 + |\lambda_2|\}$, where $\widetilde{C}_2$ is a uniform upper bound for the magnitude of all roots of p_j for $j = 1, 2, \ldots$. Then $|p_j(\lambda_2)| \le C_2^{d-1} |\lambda_2 - \sigma^{(j)}|$, where $\sigma^{(j)}$ denotes the root that converges to λ_2. Moreover, there is a constant C_1 such that $\|p_j(A_{11})^{-1}\| \le C_1$ for all j. Choosing k sufficiently large guarantees $C_1 C_2^{d-1} |\lambda_2 - \sigma^{(j)}| \le \rho$ for all $j \ge k$. Hence, there is a constant $\widetilde{C}$ such that

$$\|\hat{p}_i(A_{11})^{-1}\| \, |\hat{p}_i(\lambda_2)| \le \big(C_1 C_2^{d-1}\big)^i \prod_{j=1}^{i} |\lambda_2 - \sigma^{(j)}| \le \widetilde{C} \rho^i.$$

Combined with Theorem 3 and Remark 4, this concludes the proof. □

The following theorem incorporates the Krylov subspace acceleration benefited from choosing the deflation window size w larger than m.

Theorem 6. [illegible] 3 [illegible] $C_T \, d(\mathcal{Y}, \mathcal{K}_m(A^{\mathsf{H}}, e_n)) < 1$ [illegible] f [illegible] A

$$d\big(y, f(A)^{-\mathsf{H}} \mathcal{K}_w(A^{\mathsf{H}}, e_n)\big) \le \frac{C_T \|f(A_{11})^{-1}\| \, |f(\lambda_2)| d(y, \mathcal{K}_m(A^{\mathsf{H}}, e_n))}{1 - C_T \, d(\mathcal{Y}, \mathcal{K}_m(A^{\mathsf{H}}, e_n))} \inf_{\phi \in \mathcal{P}_{w-m}} \frac{\|\phi(A_{11})\|}{|\phi(\lambda_2)|}, \tag{30}$$

[illegible] $\mathcal{P}_{w-m}$ [illegible] $w - m$

[illegible]. As in the proof of Theorem 3, we first block diagonalize A and obtain bounds for the accordingly transformed subspaces. Let $T^{-1}AT = D$ with D as in (23) and set $\tilde{y} = T^{\mathsf{H}} y$, $\widetilde{\mathcal{Y}} = T^{\mathsf{H}} \mathcal{Y}$. Without loss of generality, we may again assume $\tilde{y} = e_n$. Moreover, we have

$$T^{\mathsf{H}} f(A)^{-\mathsf{H}} \mathcal{K}_w(A^{\mathsf{H}}, e_n) = \mathcal{K}_w(D^{\mathsf{H}}, t)$$

with $t = f(D)^{-\mathsf{H}} T^{\mathsf{H}} e_n$. It follows that

$$\begin{aligned}
d(e_n, \mathcal{K}_w(D^{\mathsf{H}}, t)) &= \inf_{v \in \mathcal{K}_w(D^{\mathsf{H}}, t)} \|e_n - v\| = \inf_{p \in \mathcal{P}_{w-1}} \|e_n - p(D^{\mathsf{H}}) t\| \\
&= \inf_{\phi \in \mathcal{P}_{w-m}} \inf_{q \in \mathcal{P}_{m-1}} \|e_n - \phi(D^{\mathsf{H}}) q(D^{\mathsf{H}}) t\| \qquad (31) \\
&= \inf_{\phi \in \mathcal{P}_{w-m}} \inf_{q \in \mathcal{P}_{m-1}} \frac{1}{|\phi(\overline{\lambda}_2)|} \|\phi(D^{\mathsf{H}})(e_n - q(D^{\mathsf{H}}) t)\| \\
&= \inf_{\phi \in \mathcal{P}_{w-m}} \inf_{v \in \mathcal{K}_m(D^{\mathsf{H}}, t)} \frac{1}{|\phi(\overline{\lambda}_2)|} \|\phi(D^{\mathsf{H}})(e_n - v)\|. \qquad (32)
\end{aligned}$$

The implicit assumption that no root of ϕ coincides with $\overline{\lambda}_2$ can be justified by the following simple argument. If $\phi(\overline{\lambda}_2)$ is zero, then $\|e_n - \phi(D^{\mathsf{H}})q(D^{\mathsf{H}})t\| \geq 1$. But since the choice $\phi \equiv q \equiv 0$ already gives $\|e_n - \phi(D^{\mathsf{H}})q(D^{\mathsf{H}})t\| = 1$, the infimum in (31) is not increased if $\phi(\overline{\lambda}_2) \neq 0$ is imposed.

The vector $\hat{v} = -\overline{f(\lambda_2)}f(D)^{-\mathsf{H}}v_1$ with v_1 defined as in (26) satisfies $\hat{v} \in \mathcal{K}_m(D^{\mathsf{H}}, t)$ and

$$e_n - \hat{v} = \begin{bmatrix} \overline{f(\lambda_2)}f(A_{11})^{-\mathsf{H}}Fe_m \\ 0 \end{bmatrix}.$$

Choosing $v = \hat{v}$ in (32) and using (28) yields

$$d(e_n, \mathcal{K}_w(D^{\mathsf{H}}, t)) \leq \frac{\|f(A_{11})^{-\mathsf{H}}\||f(\lambda_2)|}{1 - d(\widetilde{\mathcal{Y}}, \mathcal{K}_m(D^{\mathsf{H}}, t))}\, d(e_n, \widetilde{\mathcal{V}}) \inf_{\phi \in \mathcal{P}_{w-m}} \frac{\|\phi(A_{11})\|}{|\phi(\lambda_2)|}.$$

Together with Lemma 9 from the appendix, this completes the proof. ∎

Remark 4 applies analogously to Theorem 6. Comparing (30) with (22), the bound gains the factor $\inf_{\phi \in \mathcal{P}_{w-m}} \|\phi(A_{11})\|/|\phi(\lambda_2)|$. Consider a compact set $\Omega_1 \subset \mathbb{C}$ containing the eigenvalues of A_{11} and define $\kappa(\Omega_1)$ to be the smallest constant for which

$$\|g(A_{11})\| \leq \kappa(\Omega_1) \max_{z \in \Omega_1} |g(z)| \tag{33}$$

holds uniformly for every analytic function g on Ω_1. Then, trivially,

$$\inf_{\phi \in \mathcal{P}_{w-m}} \frac{\|\phi(A_{11})\|}{|\phi(\lambda_2)|} \leq \kappa(\Omega_1) \inf_{\phi \in \mathcal{P}_{w-m}} \frac{\max\{|\phi(z)| : z \in \Omega_1\}}{|\phi(\lambda_2)|}.$$

Such approximation problems play a prominent role in the convergence analysis of Krylov subspace methods for nonnormal matrices. It is clear that $\inf_{\phi \in \mathcal{P}_{w-m}} \max\{|\phi(z)| : z \in \Omega_1\}/|\phi(\lambda_2)|$ cannot be larger than 1 (choose $\phi \equiv 1$) and decays as $w - m$ becomes larger. Estimates for this decay can be found using potential theory (see [4, 12]), but this is beyond the scope of this paper.

In LAPACK 3.1 [1], the default values for the number of shifts and the size of the deflation window for $3000 \leq n < 6000$ are $m = 128$ and $w = 192$, respectively. As explained in more detail in [9], this choice was based on computational experiments with pseudo-random matrices and matrices from [2]. A good choice of these parameters is crucial for the performance of the QR algorithm. For example, choosing a larger w increases the cost of the deflation procedure (which requires $O(w^2 n)$ operations) but also results in earlier deflations and therefore in fewer multishift QR iterations (each requires $O(mn^2)$ operations). While Theorem 6 does not provide any new insight into the choice of m, it does shed some light on a beneficial choice of w. Unfortunately, the exact computation of the quantity $\inf_{\phi \in \mathcal{P}_{w-m}} \frac{\|\phi(A_{11})\|}{|\phi(\lambda_2)|}$ in (30) is infeasible, simply because A_{11} and λ_2 are available only after the Schur form of A has been computed. The results in this paper should thus be understood only as a first step toward developing a strategy that chooses w nearly optimal in each iteration. For this purpose, cheap estimates on the decay of $\inf_{\phi \in \mathcal{P}_{w-m}} \frac{\|\phi(A_{11})\|}{|\phi(\lambda_2)|}$ as w increases need to be developed, possibly using heuristics on the distribution of the eigenvalues of A_{11}.

5.2. Convergence of invariant subspaces. Not only the convergence of individual eigenvectors but also the convergence of the left invariant subspace $\mathcal{Y}$ as a whole is improved using Krylov subspaces of larger dimension. To quantify this effect, we can combine the bound (29) with results by Beattie, Embree, and Rossi [4] on the convergence of invariant subspaces in Krylov subspaces.

THEOREM 7. [illegible] $A \in \mathbb{C}^{n\times n}$ [illegible]

$$Q^{\mathsf{H}} A Q = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad A_{11} \in \mathbb{C}^{(n-m)\times(n-m)}, \quad A_{22} \in \mathbb{C}^{m\times m},$$

[illegible] $Q \in \mathbb{C}^{n\times n}$ [illegible] $\lambda(A_{11}) \cap \lambda(A_{22}) = \emptyset$ [illegible] $\mathcal{Y}$ [illegible] [illegible] A [illegible] $\lambda(A_{22})$ [illegible] [illegible] $C_T d\big(\mathcal{Y}, \mathcal{K}_m(A^{\mathsf{H}}, e_n)\big) < 1$ [illegible] $C_T = \|P_{A_{11}}\| + \sqrt{1 + \|P_{A_{11}}\|^2}$ [illegible] $P_{A_{11}}$ [illegible] [illegible] A_{11} [illegible] f [illegible] [illegible] A

(34)
$$d\big(\mathcal{Y}, f(A)^{-\mathsf{H}} \mathcal{K}_w(A^{\mathsf{H}}, e_n)\big) \le C\, \|f(A_{11})^{-1}\| \|f(A_{22})\| \inf_{\phi \in \mathcal{P}_{w-m}} \|\phi(A_{11})\| \|\phi(A_{22})^{-1}\|,$$

[illegible]

$$C = \frac{C_T^2\, d(\mathcal{Y}, \mathcal{K}_m(A^{\mathsf{H}}, e_n))}{\sqrt{1 - C_T^2\, d(\mathcal{Y}, \mathcal{K}_m(A^{\mathsf{H}}, e_n))^2}}.$$

[illegible] There is an invertible lower block triangular matrix T_1 with $\kappa(T_1) = \|P_{A_{11}}\| + \sqrt{1 + \|P_{A_{11}}\|^2}$ such that

$$(QT_1)^{-1} A (QT_1) = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} =: D.$$

Let $T = QT$ and $\widetilde{\mathcal{Y}} = T^{\mathsf{H}}\mathcal{Y}$. Then

$$T^{\mathsf{H}} f(A)^{-\mathsf{H}} \mathcal{K}_w(A^{\mathsf{H}}, e_n) = \mathcal{K}_w(D^{\mathsf{H}}, f(D)^{-\mathsf{H}} u)$$

with $u = T^{\mathsf{H}} e_n$. Partition $u = \left[\begin{smallmatrix} u_1 \\ u_2 \end{smallmatrix}\right]$ with $u_2 \in \mathbb{C}^m$. Then Theorem 3.4 in [4] states

$$d\big(\widetilde{\mathcal{Y}}, \mathcal{K}_w(D^{\mathsf{H}}, f(D)^{-\mathsf{H}} u)\big) \le C_1 \inf_{\phi \in \mathcal{P}_{w-m}} \|\phi(A_{11})\| \|\phi(A_{22})^{-1}\|,$$

where

$$C_1 = \max_{\psi \in \mathcal{P}_{m-1}} \frac{\|\psi(A_{11})^{\mathsf{H}} f(A_{11})^{-\mathsf{H}} u_1\|}{\|\psi(A_{22})^{\mathsf{H}} f(A_{22})^{-\mathsf{H}} u_2\|}.$$

Using the fact that $\psi(D)^{\mathsf{H}}$ and $f(D)^{-\mathsf{H}}$ commute, we have

$$\begin{aligned} C_1 &\le \|f(A_{11})^{-1}\| \|f(A_{22})\| \max_{\psi \in \mathcal{P}_{m-1}} \frac{\|\psi(A_{11})^{\mathsf{H}} u_1\|}{\|\psi(A_{22})^{\mathsf{H}} u_2\|} \\ &= \|f(A_{11})^{-1}\| \|f(A_{22})\| \max_{\left[\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right] \in \mathcal{K}_m(D^{\mathsf{H}}, u)} \frac{\|v_1\|}{\|v_2\|} \\ &\le \|f(A_{11})^{-1}\| \|f(A_{22})\| \frac{d\big(\widetilde{\mathcal{Y}}, \mathcal{K}_m(D^{\mathsf{H}}, u)\big)}{\sqrt{1 - d\big(\widetilde{\mathcal{Y}}, \mathcal{K}_m(D^{\mathsf{H}}, u)\big)^2}}, \end{aligned}$$

where the latter inequality follows as in the proof of Lemma 4.4 in [31]. The proof is concluded by applying Lemma 4.2 in [31] twice, relating the distances between the transformed subspaces to the distances of the original subspaces. ∎

Once again, an analogue of Remark 4 applies, showing that $C_T\, d\big(\mathcal{Y}, \mathcal{K}_m(A^{\mathsf{H}}, e_n)\big) < 1$ can be replaced by the weaker assumption $\mathcal{U} \cap \mathcal{K}_m(A^{\mathsf{H}}, e_n) = \{0\}$. Comparing (34) with the classical bound (29), the extra factor $\inf_{\phi\in\mathcal{P}_{w-m}} \|\phi(A_{11})\|\|\phi(A_{22})^{-1}\|$ is gained. If $\Omega_1, \Omega_2 \subset \mathbb{C}$ are compact sets containing $\lambda(A_{11}), \lambda(A_{22})$, respectively, then

$$\inf_{\phi\in\mathcal{P}_{w-m}} \|\phi(A_{11})\|\|\phi(A_{22})^{-1}\| \leq \kappa(\Omega_1)\kappa(\Omega_2) \min_{\phi\in\mathcal{P}_{w-m}} \frac{\max\{|\phi(z)| : z \in \Omega_1\}}{\min\{|\phi(z)| : z \in \Omega_2\}},$$

where $\kappa(\Omega_1)\kappa(\Omega_2)$ are defined as in (33). Note that the factor $\kappa(\Omega_2)$ can actually be removed (see [5, Theorem 3.3]) at the expense of having a different polynomial approximation problem. In any case, the bound can be expected to decay as $w - m$ increases; see [4, section 4] and [5, section 3] for estimates of this decay. When using Francis shifts, this means that besides the quadratically vanishing $\hat{p}_i(A_{22})$ we have an additional factor that may become very small for larger w, resulting in nearly superquadratic convergence.

Finally, let us remark that there is always a polynomial q, depending on A and f, such that $q(A) = f^{-1}(A)$. This implies an equivalence between the QR algorithm and the Arnoldi method with polynomial restarts, as discussed by Lehoucq [22]. In principle, this connection could be used to apply the bounds in [4, 5] verbatim to the QR algorithm. In contrast, Theorem 7 treats the convergence obtained from the shifts and from the Krylov subspace separately, having the advantage that it allows more insight into the benefits gained from choosing a larger deflation window.

6. Conclusions. This paper contributes to the understanding of why aggressive early deflation works so well in practice. A very intuitive explanation can be drawn from practical experiences with Krylov subspace methods for computing eigenvalues. Extracting Ritz pairs (= aggressive early deflation) is a much more effective strategy for detecting converged eigenvalues than only testing the subdiagonal entries of the Hessenberg factor in an Arnoldi decomposition (= classical deflation). The convergence bounds from section 5 provide a mathematical explanation, stating individual bounds for each converging eigenvalue and showing that the bounds are multiplied by a factor that approaches zero as the deflation window size w increases.

This paper should be seen as a first step toward developing a strategy for choosing w optimally in each QR iteration. In a serial computing environment, the current default value for w implemented in LAPACK already delivers good performance across a wide range of examples, and more sophisticated strategies might not lead to significant speedup. However, in a parallel computing environment, where aggressive early deflation will constitute a bottleneck, we expect the performance to become more sensitive with respect to the choice of w.

Finally, it is tempting to ask whether other Ritz pair extraction techniques, such as refined Ritz vectors, could be used to enhance the convergence of the QR algorithm even further. Although preliminary numerical experiments have indicated no obvious beneficial effect on the average performance of the QR algorithm, the use of these techniques in avoiding exceptional global convergence failures remains to be studied.

Appendix. This section collects two elementary facts needed in the proofs of section 5.

LEMMA 8. . . $V = \left[\begin{smallmatrix} F \\ I_m \end{smallmatrix}\right] \in \mathbb{C}^{n\times m}$ . .

$$\|e_n - Ve_m\| \le \frac{\sqrt{1+\|F\|^2}}{\sqrt{1+\|F\|^2}-\|F\|} \left\|e_n - V(V^{\mathsf{H}}V)^{-1}e_m\right\|.$$

. . . Set $r_0 = e_n - V(V^{\mathsf{H}}V)^{-1}e_m$ and $r_1 = e_n - Ve_m$. Then

$$r_0 = \begin{bmatrix} -F(I+F^{\mathsf{H}}F)^{-1} \\ I-(I+F^{\mathsf{H}}F)^{-1} \end{bmatrix} e_m = \begin{bmatrix} -(I+FF^{\mathsf{H}})^{-1}F \\ (I+F^{\mathsf{H}}F)^{-1}F^{\mathsf{H}}F \end{bmatrix} e_m$$

$$= \begin{bmatrix} (I+FF^{\mathsf{H}})^{-1} & F(I+F^{\mathsf{H}}F)^{-1} \\ -(I+F^{\mathsf{H}}F)^{-1}F^{\mathsf{H}} & (I+F^{\mathsf{H}}F)^{-1} \end{bmatrix} r_1 = (I+E)r_1,$$

where

$$E = \begin{bmatrix} -F(I+F^{\mathsf{H}}F)^{-1}F^{\mathsf{H}} & F(I+F^{\mathsf{H}}F)^{-1} \\ -(I+F^{\mathsf{H}}F)^{-1}F^{\mathsf{H}} & -F^{\mathsf{H}}(I+FF^{\mathsf{H}})^{-1}F \end{bmatrix}.$$

A singular value decomposition of F verifies $\|E\| = \|F\|/\sqrt{1+\|F\|^2}$. Thus $\|E\| < 1$ and $(I+E)$ is invertible. Finally,

$$\|r_1\| \le \|(I+E)^{-1}\|\|r_0\| \le \frac{1}{1-\|E\|}\|r_0\| = \frac{\sqrt{1+\|F\|^2}}{\sqrt{1+\|F\|^2}-\|F\|}\|r_0\|$$

concludes the proof. ∎

LEMMA 9. . $\mathcal{T},\mathcal{U}$ . . . . . P . . . . . .
. . $s \in \mathbb{C}^n$, $d(s,\mathcal{T}) \le \alpha\, d(s,\mathcal{U})$. . . . $d(Ps,P\mathcal{T}) \le \alpha\|P\|\|P^{-1}\|\, d(Ps,P\mathcal{U})$
. . . . Without loss of generality, we may assume $\|s\| = 1$. Then

$$\begin{aligned} d(Ps,P\mathcal{T}) &= \inf_{t\in\mathcal{T}} \frac{\|Ps-Pt\|}{\|Ps\|} \le \frac{\|P\|}{\|Ps\|} \inf_{t\in\mathcal{T}} \|s-t\| \\ &\le \alpha \frac{\|P\|}{\|Ps\|} \inf_{u\in\mathcal{U}} \|s-u\| \le \alpha\,\|P\|\|P^{-1}\| \inf_{u\in\mathcal{U}} \frac{\|Ps-Pu\|}{\|Ps\|} \\ &= \alpha\,\|P\|\|P^{-1}\|\, d(Ps,P\mathcal{U}). \end{aligned}$$ ∎

Acknowledgments. The author is very grateful for illuminating discussions with Ralph Byers, who first pointed out the close relationship between aggressive early deflation and the Krylov–Schur algorithm. He also thanks Mark Embree and the referees for carefully reading the paper and providing helpful suggestions.

REFERENCES

[1] E. ANDERSON, Z. BAI, C. H. BISCHOF, S. BLACKFORD, J. W. DEMMEL, J. J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. C. SORENSEN, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, PA, 1999.

[2] Z. BAI, D. DAY, J. W. DEMMEL, AND J. J. DONGARRA, *A test matrix collection for non-Hermitian eigenvalue problems (release 1.0)*, Technical Report CS-97-355, Department of Computer Science, University of Tennessee, Knoxville, TN, 1997; also available online from http://math.nist.gov/MatrixMarket.

[3] F. L. Bauer, *Das Verfahren der Treppeniteration und verwandte Verfahren zur Lösung algebraischer Eigenwertprobleme*, Z. Angew. Math. Phys., 8 (1957), pp. 214–235.
[4] C. A. Beattie, M. Embree, and J. Rossi, *Convergence of restarted Krylov subspaces to invariant subspaces*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1074–1109.
[5] C. A. Beattie, M. Embree, and D. C. Sorensen, *Convergence of polynomial restart Krylov methods for eigenvalue computations*, SIAM Rev., 47 (2005), pp. 492–515.
[6] K. Braman, R. Byers, and R. Mathias, *The multishift QR algorithm.* I. *Maintaining well-focused shifts and level* 3 *performance*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 929–947.
[7] K. Braman, R. Byers, and R. Mathias, *The multishift QR algorithm.* II. *Aggressive early deflation*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 948–973.
[8] H. J. Buurema, *A Geometric Proof of Convergence for the QR Method*, Doctoral dissertation, University of Groningen, Rijksuniversiteit te Groningen, Groningen, The Netherlands, 1970.
[9] R. Byers, *LAPACK* 3.1 *xHSEQR: Tuning and implementation notes on the small bulge multi-shift QR algorithm with aggressive early deflation*, LAPACK Working Note 187, 2007.
[10] G. Chen and Z. Jia, *A reverse order implicit Q-theorem and the Arnoldi process*, J. Comput. Math., 20 (2002), pp. 519–524.
[11] J. W. Demmel, *Computing stable eigendecompositions of matrices*, Linear Algebra Appl., 79 (1986), pp. 163–193.
[12] T. A. Driscoll, K.-C. Toh, and L. N. Trefethen, *From potential theory to matrix iterations in six steps*, SIAM Rev., 40 (1998), pp. 547–578.
[13] J. G. F. Francis, *The QR transformation* I: *A unitary analogue to the LR transformation*, Comput. J., 4 (1961), pp. 265–271.
[14] J. G. F. Francis, *The QR transformation* II, Comput. J., (1962), pp. 332–345.
[15] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[16] Z. Jia, *Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems*, Linear Algebra Appl., 259 (1997), pp. 1–23.
[17] Z. Jia and G. W. Stewart, *An analysis of the Rayleigh-Ritz method for approximating eigenspaces*, Math. Comp., 70 (2001), pp. 637–647.
[18] D. Kressner, *Numerical Methods and Software for General and Structured Eigenvalue Problems*, Ph.D. thesis, TU Berlin, Institut für Mathematik, Berlin, Germany, 2004.
[19] D. Kressner, *Deflation in Krylov subspace methods and distance to uncontrollability*, Ann. Univ. Ferrara Sez. VII Sci. Mat., 53 (2007), pp. 309–318.
[20] V. N. Kublanovskaya, *On some algorithms for the solution of the complete eigenvalue problem*, Zhurnal Vychislitelnoi Matematiki i Matematicheskoi Fiziki, 1 (1961), pp. 555–570.
[21] B. Lang, *Effiziente Orthogonaltransformationen bei der Eigen- und Singulärwertzerlegung*, Habilitationsschrift, Fachbereich Mathematik, Bergische Universität GH Wuppertal, Wuppertal, Germany, 1997.
[22] R. B. Lehoucq, *Implicitly restarted Arnoldi methods and subspace iteration*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 551–562.
[23] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK users' guide*, Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods, SIAM, Philadelphia, PA, 1999.
[24] B. N. Parlett and W. G. Poole, Jr., *A geometric theory for the* QR, LU *and power iterations*, SIAM J. Numer. Anal., 10 (1973), pp. 389–412.
[25] Y. Saad, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl., 34 (1980), pp. 269–295.
[26] G. W. Stewart, *Matrix Algorithms*, Vol. II, Eigensystems, SIAM, Philadelphia, PA, 2001.
[27] G. W. Stewart, *A Krylov–Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 601–614.
[28] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
[29] D. S. Watkins, *Understanding the* QR *algorithm*, SIAM Rev., 24 (1982), pp. 427–440.
[30] D. S. Watkins, *The QR algorithm revisited*, SIAM Rev., 50 (2008), pp. 133–145.
[31] D. S. Watkins and L. Elsner, *Convergence of algorithms of decomposition type for the eigenvalue problem*, Linear Algebra Appl., 143 (1991), pp. 19–47.
[32] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.
[33] J.-P. M. Zemke, *Hessenberg eigenvalue-eigenmatrix relations*, Linear Algebra Appl., 414 (2006), pp. 589–606.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 822–843

A NEW SCALING FOR NEWTON'S ITERATION FOR THE POLAR DECOMPOSITION AND ITS BACKWARD STABILITY*

RALPH BYERS† AND HONGGUO XU‡

Abstract. We propose a scaling scheme for Newton's iteration for calculating the polar decomposition. The scaling factors are generated by a simple scalar iteration in which the initial value depends only on estimates of the extreme singular values of the original matrix, which can, for example, be the Frobenius norms of the matrix and its inverse. In exact arithmetic, for matrices with condition number no greater than 10^{16}, with this scaling scheme no more than 9 iterations are needed for convergence to the unitary polar factor with a convergence tolerance roughly equal to 10^{-16}. It is proved that if matrix inverses computed in finite precision arithmetic satisfy a backward-forward error model, then the numerical method is backward stable. It is also proved that Newton's method with Higham's scaling or with Frobenius norm scaling is backward stable.

Key words. matrix sign function, polar decomposition, singular value decomposition (SVD), Newton's method, numerical stability, scaling

AMS subject classifications. 65F05, 65G05

DOI. 10.1137/070699895

1. Introduction.

Every matrix $A \in \mathbb{C}^{n\times n}$ has a polar decomposition $A = QH$, where $H = H^* \in \mathbb{C}^{n\times n}$ is Hermitian positive semidefinite and $Q \in \mathbb{C}^{n\times n}$ is unitary, i.e., $Q^*Q = I$. The polar decomposition is unique with positive definite symmetric factor H iff A is nonsingular. Its applications include unitary approximation and distance calculations [8, 9, 12]. The polar decomposition generalizes to rectangular matrices; see, for example, [15]. We consider only the square matrix case here, because numerical methods for computing the polar decomposition typically begin by reducing the problem down to the square matrix case using, for example, a QR factorization [5, 8]. (An algorithm that works directly with rectangular matrices appears in [6].)

The polar decomposition may be easily constructed from a singular value decomposition (SVD) of A. However, the SVD is a substantial calculation that displays much more of the structure of A than does the polar decomposition. Constructing the polar decomposition from the SVD destroys this extra information and wastes the arithmetic work used to compute it. It is intuitively more appealing to use the polar decomposition as a preliminary step in the computation of the SVD as in [12].

When A is nonsingular, one way to compute the polar decomposition is through Newton's iteration:

$$Q_{k+1} = \frac{1}{2}\left(\zeta_k Q_k + (\zeta_k Q_k)^{-*}\right), \qquad Q_0 = A, \tag{1.1}$$

where $\zeta_k = \zeta(Q_k) > 0$ is a positive scalar function of Q_k chosen to accelerate convergence [8]. Each iterate Q_k has polar decomposition $Q_k = QH_k$, where Q is

*Received by the editors August 13, 2007; accepted for publication (in revised form) by N. J. Higham April 24, 2008; published electronically September 4, 2008.
http://www.siam.org/journals/simax/30-2/69989.html

†This author is deceased. Former address: Department of Mathematics, University of Kansas, Lawrence, KS 66045. This material was based upon work partially supported by the University of Kansas General Research Fund allocations 2301062-003 and 2301054-003 and by the NSF awards 0098150, 0112375, and 9977352.

‡Department of Mathematics, University of Kansas, Lawrence, KS 66045 (xu@math.ku.edu). This author was partially supported by NSF grant EPS-9874732 with matching support from the State of Kansas and the University of Kansas General Research Fund allocation 2301717-003.

the unitary polar factor of A, $H_0 = H$ is the Hermitian polar factor of A, and $H_{k+1} = (\zeta_k H_k + (\zeta_k H_k)^{-1})/2$, $k \geq 0$. For appropriately chosen acceleration parameters ζ_k, $\lim_{k\to\infty} H_k = I$. Hence, the unitary polar factor is $Q = \lim_{k\to\infty} Q_k$, and the Hermitian polar factor is $H = \lim_{k\to\infty} Q_k^* A$.

Iteration (1.1) was first proposed in [8] and studied further in [5, 6, 14]. It is called "Newton's iteration" because it can be derived from Newton's method applied to the equation $X^*X = I$. It is closely related to Newton's iteration for the matrix sign function [17, 22].

Simplicity is an attractive feature of (1.1). Apart from the computation of ζ_k, each iteration needs only one matrix inversion and one matrix-matrix addition. The simplicity allows implementations of (1.1) to take advantage of the hierarchical memory and parallelism [1, 2, 13]. Many authors have studied choices of the acceleration parameters ζ_k [3, 4, 8, 16, 17, 22]. If $\zeta_k \equiv 1$, then the iterates Q_k converge quadratically to the unitary polar factor Q [8]. Convergence is also quadratic if $\zeta = \zeta(U)$ is a smooth function of $U \in \mathbb{C}^{n\times n}$ and $\zeta(U) = 1$ whenever U is unitary.

The choice

$$\zeta_k^{(2)} = \sqrt{\frac{\|Q_k^{-1}\|_2}{\|Q_k\|_2}}, \tag{1.2}$$

where $\|\cdot\|_2$ is the spectral norm is proposed in [8]. This scale factor is optimal in the sense that, given Q_k, (1.2) minimizes the next error $\|Q_{k+1} - Q\|_2$. With this scale factor, for the matrices Q_k generated by (1.1), the error sequence $\|Q_k - Q\|_2$ converges monotonically to zero. Unfortunately, to determine the scale factor (1.2), one needs to compute two extreme singular values of Q_k at each iteration. In order to preserve the rapid convergence of (1.1) with scaling (1.2), the highly accurate values of these extreme singular values are required to guarantee $\zeta_k^{(2)} \to 1$. This is expensive enough to make scale factor (1.2) unattractive.

To save the cost of computing the extreme singular values, one might approximate (1.2). A commonly used scale factor is the $(1,\infty)$-scaling

$$\zeta_k^{(1,\infty)} = \left(\frac{\|Q_k^{-1}\|_1 \|Q_k^{-1}\|_\infty}{\|Q_k\|_1 \|Q_k\|_\infty}\right)^{\frac{1}{4}}, \tag{1.3}$$

(where $\|\cdot\|_1$ and $\|\cdot\|_\infty$ are the 1-norm and ∞-norm, respectively) which was proposed by Higham in [8]. The factor $\zeta_k^{(1,\infty)}$ is within a constant factor of $\zeta_k^{(2)}$. It adds a negligible amount of arithmetic work compared to the cost of Q_k^{-1}, which is needed at each iteration anyway.

The scale factor

$$\zeta_k^{(F)} = \|Q_k^{-1}\|_F^{1/2} \|Q_k\|_F^{-1/2}, \tag{1.4}$$

(where $\|\cdot\|_F$ is the Frobenius norm) is discussed in [5, 8, 16]. It can also be computed at a negligible cost. It is optimal in the sense that, given Q_k, it minimizes $\|Q_{k+1}\|_F$ and causes the sequence $\|Q_k\|_F$ to converge monotonically [5].

Another relatively inexpensive scale factor is [4]

$$\zeta_k^{(d)} = |\det(Q_k)|^{-1/n}. \tag{1.5}$$

The complex modulus of the determinant is very inexpensively obtained from the same matrix factorization used to calculate Q_k^{-1}. This scaling is optimal in the sense

that, for a given iterate Q_k, it minimizes $D(Q_{k+1}) = \sum_{j=1}^{n} (\ln(\sigma_j^{(k+1)}))^2$, where $\sigma_j^{(k+1)}$ is the jth singular value of Q_{k+1}. The function $D(Q_{k+1})$ is a measure of the departure of Q_{k+1} from the unitary matrices.

This paper considers the [illegible] scaling strategy

$$\zeta_0 = 1/\sqrt{ab}, \quad \zeta_1 = \sqrt{\frac{2\sqrt{ab}}{a+b}}, \quad \zeta_k = 1/\sqrt{\rho(\zeta_{k-1})}, \quad k = 2, 3, \ldots, \tag{1.6}$$

where $\rho(x) = (x + x^{-1})/2$ and a and b are any numbers such that $0 < a \le \|A^{-1}\|_2^{-1} \le \|A\|_2 \le b$. Apart from estimating the extreme singular values of the initial matrix $Q_0 = A$, the scale factor costs only several floating point operations per iteration. Moreover, only the rough estimates of $\|A\|_2$ and $\|A^{-1}\|_2^{-1}$ are needed. One may simply choose $a = \|A^{-1}\|_F^{-1}$ and $b = \|A\|_F$. From Table 2.1 with such choices for any matrices with condition number no greater than 10^{16} and size no greater than 10^{11}, at most nine iterations of (1.1) with scaling (1.6) are necessary to approximate the unitary polar factor Q to within 2-norm distance less than 10^{-16}.

We show below that, in the presence of rounding error, (1.1) with (1.6) is numerically stable assuming that matrix inverses are calculated with small forward-backward error. This is the case, for example, when matrix inverses are computed using the bidiagonal reduction [7, p. 252]. We also prove the numerical stability of Newton's iteration with any of the scalings (1.2)–(1.4).

Commenting on an early draft of this paper, Zięta̧k pointed out that the suboptimal (quasi-optimal) scaling parameters were discovered independently by Kiełbasiński but not published in the open literature. They were presented by Zięta̧k at the 1999 Householder meeting at Whistler and the 1999 ILAS conference at Barcelona. In section 5 of their recent paper [19], Kiełbasiński, Zieliński, and Zięta̧k mention the quasi-optimal scaling parameters. In [18], these authors gave an error analysis of Higham's method [8] with the same mixed backward-forward stability assumption for matrix inversion. They gave numerical experiments in [23].

In the following $A \in \mathbb{C}^{n\times n}$ is always nonsingular. $A = U\Sigma V^*$ is the SVD of A, where U and V are unitary, $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_n)$ is diagonal, and $\sigma_1 \ge \cdots \ge \sigma_n \ge 0$ are the singular values of A. The set of the singular values is denoted by $\sigma(A)$. The condition number with respect to the spectral norm of A is denoted by $\kappa_2(A) = \sigma_1/\sigma_n$. Following [7, p. 18], a flop is the computational work of a floating point addition, subtraction, multiplication, or division together with the associated subscripting and indexing overhead. It takes two flops to execute the Fortran statement `A(I,J) = A(I,J) + C*A(K,J)`.

2. Scaling and convergence. Let $A \in \mathbb{C}^{n\times n}$ be nonsingular, with the SVD $A = U\Sigma V^*$, with $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$. Each Newton iterate Q_k in (1.1) has the SVD $Q_k = U\Sigma_k V^*$, where

$$\Sigma_{k+1} = \left(\zeta_k \Sigma_k + (\zeta_k \Sigma_k)^{-1}\right)/2, \qquad \Sigma_0 = \Sigma. \tag{2.1}$$

In particular, Q_k has singular values $\sigma_1^{(k)}$, $\sigma_2^{(k)}$, $\ldots$ $\sigma_n^{(k)}$ (in no particular order when $k > 0$) that obey

$$\sigma_j^{(0)} = \sigma_j, \quad \sigma_j^{(k+1)} = \frac{1}{2}\left(\zeta_k \sigma_j^{(k)} + \frac{1}{\zeta_k \sigma_j^{(k)}}\right) = \rho(\zeta_k \sigma_j^{(k)}), \quad k = 0, 1, 2, \ldots, \tag{2.2}$$

where $\rho(x) = (x + x^{-1})/2$. For appropriately chosen ζ_k, $\lim_{k\to\infty} \sigma_j^{(k)} = 1$, and $\lim_{k\to\infty} Q_k = UV^* = Q$. Consequently, the convergence properties of (1.1) derive directly from the n scalar sequences $\sigma_j^{(k)}$ determined by (2.2). To attain good convergence behavior in (1.1), the acceleration parameters ζ_k must interact well with $\rho(x) = (x + x^{-1})/2$.

The following two lemmas list some easily verified elementary properties of $\rho(x) = (x + x^{-1})/2$.

LEMMA 2.1. [illegible] $x > 0$ [illegible]

1. $\rho(\frac{1}{x}) = \rho(x)$
2. $1 \le \rho(x) \le \max(x, x^{-1})$ [illegible] $x = 1$
3. $\rho(x)$ [illegible] $x \in (0, 1]$ [illegible] $x \in [1, \infty)$

LEMMA 2.2. [illegible] $0 < a \le b$ [illegible] $\alpha_\zeta = \max\{(\zeta a)^{-1}, \zeta b\}$ [illegible] $\zeta_{opt} = (ab)^{-1/2}$ [illegible]

1. [illegible] $\zeta > 0$, $1 \le \max_{a<x<b} \rho(\zeta x) = \rho(\alpha_\zeta)$ [illegible] $1 = \max_{a<x<b} \rho(\zeta x)$. $\alpha_\zeta = 1$
2. [illegible] $\zeta > 0$, $1 \le \min_{a<x<b} \rho(\zeta x)$ [illegible] $1 = \min_{a<x<b} \rho(\zeta x)$. $\zeta a \le 1 \le \zeta b$
3. $\min_{\zeta>0} \alpha_\zeta = \alpha_{\zeta_{opt}} = \sqrt{b/a}$ [illegible] $\zeta = \zeta_{opt}$ [illegible]
4. $\min_{\zeta>0} \max_{a\le x\le b} \rho(\zeta x) = \min_{\zeta>0} \rho(\alpha_\zeta) = \rho(\alpha_{\zeta_{opt}}) = \rho(\sqrt{b/a})$

In the following, for ease of notation let $\tau(x)$ be the function

$$\tau(x) = \rho(\sqrt{x}) = \frac{1}{2}\left(\sqrt{x} + \frac{1}{\sqrt{x}}\right). \tag{2.3}$$

The k-fold composition of $\tau(x)$ with itself is written $\tau^k(x)$, i.e., $\tau^0(x) = x$, $\tau^1(x) = \tau(x)$, and for $k > 1$, $\tau^{k+1}(x) = \tau(\tau^k(x))$. Similarly $\rho^k(x)$ is the k-fold composition of $\rho(x) = (x + x^{-1})/2$ with itself.

Suppose that $0 < a \le \sigma_n \le \sigma_1 \le b$. Consider the sequence of intervals generated by Newton's iteration: $[a_0, b_0] = [a, b]$, $[a_1, b_1] = \rho(\zeta_0[a_0, b_0])$, $[a_2, b_2] = \rho(\zeta_1[a_1, b_1])$, ... It follows from (2.2) that $\sigma_j^{(k)} \in [a_k, b_k]$, $j = 1, 2, \ldots, n$, $k = 0, 1, \ldots$. Note that $\min_{x>0} \rho(x) = 1$, so for $k \ge 1$, $[a_k, b_k] \subseteq [1, b_k]$ and

$$1 \le \sigma_j^{(k)} \le b_k. \tag{2.4}$$

It is intuitively satisfying to choose the sequence of acceleration parameters ζ_k in (1.1) to minimize the sequence b_k.

From Lemma 2.2, the initial optimal scaling factor is $\zeta_0 = (ab)^{-1/2}$. The initial interval is scaled to be $\zeta_0[a, b] = [\sqrt{a/b}, \sqrt{b/a}]$ which contains 1. The next interval is

$$[a_1, b_1] = \rho(\zeta_0[a, b]) = [1, \rho(\sqrt{b/a})] = [1, \tau(b/a)],$$

where $\tau(x)$ is given by (2.3). The left endpoint is $a_1 = 1$, so the optimal scaling factor for the next iteration is $\zeta_1 = b_1^{-1/2} = 1/\sqrt{\tau(b/a)}$. The next interval is

$$[a_2, b_2] = \rho(\zeta_1[a_1, b_1]) = [1, \rho(\sqrt{\tau(b/a)})] = [1, \tau^2(b/a)].$$

An easy induction shows that the sequence of intervals is $[a_0, b_0] = [a, b]$ and for $k \ge 1$, $[a_k, b_k] = [1, \tau^k(b/a)]$, and the sequence of optimal scaling factors is

$$\zeta_0 = 1/\sqrt{ab}, \quad \zeta_k = 1/\sqrt{\tau^k(b/a)}, \quad k = 1, 2, 3, \ldots, \tag{2.5}$$

which is equivalent to (1.6).

Since $\tau(x) = \rho(\sqrt{x}) \le \rho(x)$ for $x \ge 1$,

$$\tau(b/a) \le \rho(b/a).$$

By induction we have

$$\tau^k(b/a) \le \rho^k(b/a)$$

for all $k \ge 1$. The sequence $b/a, \rho(b/a), \rho^2(b/a), \ldots$, is generated by Newton's iteration $x_{k+1} = \rho(x_k)$, with $x_0 = b/a$. It converges to 1 quadratically. Obviously $\tau^k(b/a) \ge 1$ for $k \ge 0$. So $b/a, \tau(b/a), \tau^2(b/a), \ldots$ also converges to 1 at least quadratically. It is not difficult to show that $1 \le \tau(x) \le x$ for any $x \ge 1$. We have

$$b_k = \tau^k(b/a) = \tau(\tau^{k-1}(b/a)) \le \tau^{k-1}(b/a) = b_{k-1}.$$

Hence, after the first step, the sequence of intervals satisfies

$$[a_1, b_1] \supseteq [a_2, b_2] \supseteq \cdots \supseteq [a_k, b_k] \supseteq \ldots,$$

and it converges to the single point 1 quadratically. (Note $a_k = 1$ for all $k \ge 1$.) The initial interval, $[a_0, b_0] = [a, b]$ is an exception, because, in general, $[a, b] \not\supseteq [1, \tau(b/a)]$.

Based on this fact and (2.4), the convergence properties of (1.1) with (1.6) are clear, and we summarize them in the following theorem.

THEOREM 2.3.

$$0 < a \le \|A^{-1}\|_2^{-1} \le \|A\|_2 \le b \tag{2.6}$$

Q_k (1.1) (1.6)

$$\|Q_k - Q\| \le \tau^k(b/a) - 1 \le \rho^k(b/a) - 1, \quad k = 1, 2, \ldots. \tag{2.7}$$

In fact the convergence properties are highly satisfactory even when b/a is large. Table 2.1 uses Theorem 2.3 to list the number of Newton's iteration (1.1) with scaling (1.6) (and exact arithmetic) required to guarantee selected absolute errors $\delta > \|Q_k - Q\|_2$ and values of b/a. The table demonstrates that Newton's iteration (1.1) with scaling (1.6) typically needs no more than nine iterations to attain typical floating point precision accuracy. The table also demonstrates that convergence is insensitive to the choice of a and b—widely differing values of b/a need similar numbers of iterations to attain similar accuracy. In particular the easy-to-compute choices $a = \|A^{-1}\|_F^{-1}$ and $b = \|A\|_F$ satisfy (2.6) and are unlikely to lead to even one more iteration than the optimum choices of $a = \|A^{-1}\|_2^{-1}$ and $b = \|A\|_2$, particularly for ill-conditioned matrices. For instance, for any $A \in \mathbb{C}^{n\times n}$ with $\kappa_2(A) = 10^{16}$, for $a = \|A^{-1}\|_F^{-1}$ and $b = \|A\|_F$, we have $b/a \le n\kappa(A) = n10^{16}$. Then $b/a \le 10^{27}$ for any $n \le 10^{11}$, and the number of iterations is 9, the same as with the optimum choices.

In Theorem 2.3, smaller values of b/a give smaller values of $\tau^k(b/a)$ and hence better error bounds. Inequality (2.6) implies that $b/a \ge \kappa_2(A) = \|A^{-1}\|_2 \|A\|_2$, and equality can be achieved only with $a = \|A^{-1}\|_2^{-1} = \sigma_n$ and $b = \|A\|_2 = \sigma_1$. With $a = \sigma_n$ and $b = \sigma_1$, the scaling factors (1.6) are

$$\zeta_0 = 1/\sqrt{\sigma_1\sigma_n}, \qquad \zeta_k = 1/\sqrt{\tau^k(\kappa_2(A))}, \quad (k \ge 1), \tag{2.8}$$

and the corresponding intervals are $[\sigma_n, \sigma_1]$, and $[1, \tau^k(\kappa_2(A))]$ for $k \ge 1$. Let Σ_k be the matrices generated by (2.1), with $a = \sigma_n$ and $b = \sigma_1$. It is easy to verify that in

TABLE 2.1
The number of Newton iterations (1.1) *with scaling* (1.6) *(and exact arithmetic) required to guarantee absolute error* $\|Q_k - Q\|_2 < \delta$ *for selected values of* δ *and* b/a *such that* $0 < a \le \|A^{-1}\|_2^{-1} \le \|A\|_2 \le b$. *See Theorem* 2.3.

$\delta \setminus b/a$	10	10^5	10^{10}	10^{15}	10^{20}	10^{25}	10^{27}
10^{-1}	2	4	5	6	6	6	6
10^{-4}	4	5	6	7	7	8	8
10^{-7}	4	6	7	8	8	8	8
10^{-10}	5	7	7	8	8	9	9
10^{-13}	5	7	8	8	9	9	9
10^{-16}	5	7	8	9	9	9	9
10^{-19}	6	7	8	9	9	10	10

this case, the right endpoint of the kth interval is a singular value of Σ_k. This in turn implies that inequality (2.7) is an equality, i.e.,

$$\|Q_k - Q\|_2 = \|\Sigma_k - I\|_2 = \tau^k(\kappa_2(A)) - 1.$$

The number sequence b_k was also derived in [16] in order to show the convergence behavior of Newton's method with the optimal scale factors. It is shown that when $a = \sigma_n$ and $b = \sigma_1$, for Q_k generated with $\zeta_{k-1}^{(2)}$ defined in (1.2), one has $\|Q_k\|_2 \le b_k$ [16, 11]. Due to this fact we call ζ_k defined in (2.5) [illegible] scale factors. Note that b_k is derived based on different interpretations here. It is the right endpoint of the kth interval generated by applying Newton's iteration to the initial interval $[a, b]$. For this interval iteration, ζ_k is the scale factor that minimizes $b_{k+1} - 1$ (i.e., it makes $[a_{k+1}, b_{k+1}]$ as close to 1 as possible).

3. The algorithm. The Newton's method (1.1) with scaling scheme (1.6) is implemented by the following algorithm.

ALGORITHM 3.1 (Newton's method (1.1) with scaling (1.6)).
Input: Nonsingular matrix $A \in \mathbb{C}^{n\times n}$ and a stopping criterion $\delta > 0$.
Output: The polar decomposition $A = QH$.

Step 0: **a.** Set $Q_0 = A$; Compute Q_0^{-*}
b. Choose $a \le \|Q_0^{-1}\|_2^{-1}$ and $b \ge \|Q_0\|_2$; $\zeta_0 = 1/\sqrt{ab}$
c. Set $k = 0$
Step 1: While $\|Q_k - Q_k^{-*}\|_F \ge \delta$
a. $Q_{k+1} = (\zeta_k Q_k + \zeta_k^{-1} Q_k^{-*})/2$
b.
If $k = 0$, $\zeta_1 = \sqrt{\frac{2}{\sqrt{b/a}+\sqrt{a/b}}}$
Else $\zeta_{k+1} = \sqrt{\frac{2}{\zeta_k^{-1}+\zeta_k}}$
End if
c. Compute Q_{k+1}^{-*}
d. $k = k + 1$
End while
Step 2: $Q = (Q_k + Q_k^{-*})/2$; $H = \frac{1}{2}(Q^*A + (Q^*A)^*)$

Here are some remarks.

1. The matrix $A^{-1} = Q_0^{-1}$ needs to be computed in the first iteration anyway. Hence, power iterations on A and A^{-1} may be used evaluate the extreme singular values σ_1 and σ_n^{-1}, respectively, using only $O(n^2)$ extra flops per iteration. These

estimates may then serve as the suboptimal scaling factors b and a^{-1}, respectively. Since highly accurate estimates of σ_1 and σ_n are unnecessary, a few power iterations should suffice. Alternatively, $\|A\|_F$ and $\|A^{-1}\|_F^{-1}$ may be used for b and a.

2. The stopping criterion $\|Q_k - Q_k^{-*}\|_F < \delta$ is essentially equivalent to $\|Q_{k+1} - Q_k\|_F < \delta$, which is used in [11, section 8.9]. This follows from the fact that when $\zeta_k \approx 1$ (which is usually the case for a small δ),

$$Q_{k+1} - Q_k = \frac{1}{2}(\zeta_k^{-1} Q_k^{-*} - (2 - \zeta_k) Q_k) \approx \frac{1}{2}(Q_k^{-*} - Q_k).$$

In practice, in order for the computed Q_k to be within $O(\varepsilon)$ of a unitary matrix, where ε is the machine epsilon, it is sufficient to choose $\delta = O(\sqrt{\varepsilon})$. See (4.16).

3. Commonly the matrix inversion method used in the algorithm is an LU factorization-based method such as the Gaussian elimination with partial pivoting or complete pivoting [8]. Such an inversion method usually works well in practice [18]. In order to guarantee the algorithm to be numerically backward stable, one may use the more expensive bidiagonal reduction-based matrix inversion method provided in Appendix A.1. So the computed matrix inverses satisfy the backward-forward error model. See Assumption 4.1 in section 4 below.

4. Estimating σ_1 and σ_n usually uses $O(n^2)$ flops. Each iteration uses $2n^3$ flops for the matrix inverse by an LU factorization-based method and $O(n^2)$ flops for matrix addition. Computing H uses $2n^3$ flops. If p is the number of iterations for convergence, then the algorithm uses a total of roughly $2(p+1)n^3$ flops [8]. When $p = 9$ it is about $20n^3$ flops, which is less than the QR-like SVD method (which takes $22n^3$ to $26n^3$ flops for the SVD and $4n^3$ for Q and H). If the bidiagonal reduction-based matrix inversion method is used, the total cost will be $2(3p + 1)n^3$ flops.

5. In order to reduce the cost while maintaining numerical stability, one may first use the bidiagonal reduction-based method for a few iterations. When $\kappa_2(Q_k)$ is not too large, say 100, one shifts to an LU factorization-based inversion method for the subsequent iterations. The matrix inverses essentially satisfy the backward-forward error model in the latter case [10, section 14]. Also, it takes only a few iterations for the condition number to drop below 100. In the case when $\kappa_2(A) = 10^{-16}$, with $a = \sigma_n$ and $b = \sigma_1$, then $\|Q_3\|_2 = \tau^3(10^{-16}) \approx 42$. Since this is usually the worst case in practice, the bidiagonal reduction-based method is required in no more than 3 iterations. With this strategy, the maximum cost (with $p = 9$) is $3 \cdot 6n^3 + (9 - 3) \cdot 2n^3 + 2n^3 = 32n^3$ flops.

6. Although the cost for computing the scale factors (1.3)–(1.5) is negligible in Newton's iteration, computing the suboptimal scale factors is essentially costless. Also, the use of suboptimal scaling simplifies the algorithm, since the "shifting scale factor to 1" strategy, which is used for the $(1, \infty)$-scaling ([8]), is not needed. Finally, with suboptimal scaling, in general, the number of iterations is no greater than 9, and it can be obtained by simply computing $\tau^k(b/a) - 1$. It is still not clear how to predict the number of iterations with other scalings, although in practice it is observed that the $(1, \infty)$-scaling and the suboptimal scaling essentially have the same convergence rate.

4. Stability and rounding error analysis. In this section, a first order error analysis establishes that Newton's method (1.1) with scaling (1.6) can be implemented in a backward stable way. The same conclusion is drawn for the scalings (1.2), (1.3), and (1.4). In outline, the approach is to estimate the residual $\|A - \widehat{Q}\widehat{H}\|_2$ for the rounding-error-perturbed unitary factor $\widehat{Q}$ and Hermitian factor $\widehat{H}$ produced by finite

precision arithmetic in the algorithm in section 3. The method here is to first estimate the forward errors $\widehat{Q} - Q$ and $\widehat{H} - H$ and then use them to estimate the residual.

For the error analysis, we employ the standard model of floating point arithmetic with machine epsilon ε [10, section 2.2].

We also need the following assumptions.

ASSUMPTION 4.1. If a nonsingular matrix $A \in \mathbb{C}^{n\times n}$ is inverted using finite precision arithmetic with machine epsilon ε to obtain a "computed inverse" X, then

$$X = (A + E)^{-1} + F,$$

where E, $F \in \mathbb{C}^{n\times n}$ are perturbation matrices satisfying

$$\|E\|_2 \le c_1(n)\varepsilon\|A\|_2, \qquad \|F\|_2 \le c_2(n)\varepsilon\|A^{-1}\|_2,$$

and $c_i(n)$ $(i = 1, 2)$ are some low-degree polynomials of n.

In Newton's method it is typical to use Gaussian elimination with partial or complete pivoting for computing matrix inverses. Although it works well in practice, the computed matrix inverses may not satisfy Assumption 4.1 [10, section 14.1]. We show in Appendix A.1 that Assumption 4.1 is satisfied by a matrix inversion algorithm that uses the bidiagonal reduction method.

ASSUMPTION 4.2.

$$c_3(n)\kappa_2(A)\varepsilon < 1,$$

where $c_3(n)$ is a low-degree polynomial of n.

Note that $1/\kappa_2(A)$ is the measure of the relative distance of a nonsingular A to the nearest singular matrices [7, p. 73], i.e.,

$$\frac{1}{\kappa_2(A)} = \min_{\det(A+E)=0} \frac{\|E\|_2}{\|A\|_2}.$$

If such a condition doesn't hold, then matrices like $A + E$ in Assumption 4.1 can be singular. So this is a condition about the numerical nonsingularity of A. It is essential in the subsequent first order error analysis, although it won't be explicitly stated.

We now begin the error analysis. In practice, rounding errors perturb Newton's method recurrence (1.1). Under Assumptions 4.1, if $\widehat{Q}_k$ is the computed version of Q_k, then

$$\begin{aligned}
\widehat{Q}_{k+1} &= \frac{\zeta_k}{2}\widehat{Q}_k + F_{k,1} + \frac{1}{2\zeta_k}\left((\widehat{Q}_k + F_{k,2})^{-*} + F_{k,3}\right) \\
&= \frac{\zeta_k}{2}(\widehat{Q}_k + F_{k,2}) + \frac{1}{2\zeta_k}(\widehat{Q}_k + F_{k,2})^{-*} + \left(F_{k,1} + \frac{1}{2\zeta_k}F_{k,3} - \frac{\zeta_k}{2}F_{k,2}\right) \\
&=: \frac{\zeta_k}{2}(\widehat{Q}_k + F_{kb}) + \frac{1}{2\zeta_k}(\widehat{Q}_k + F_{kb})^{-*} + F_{kf}, \qquad (4.1)
\end{aligned}$$

where $F_{kb} = F_{k,2}$ and $F_{kf} = (F_{k,1} + \frac{1}{2\zeta_k}F_{k,3} - \frac{\zeta_k}{2}F_{k,2})$. The perturbation matrix $F_{k,1}$ represents rounding errors introduced by floating point matrix addition and scalar multiplication, and the perturbation matrices $F_{k,2}$ and $F_{k,3}$ represent rounding errors

introduced by matrix inversion under Assumption 4.1. The F's obey the bounds

$$\begin{aligned}
\|F_{k,1}\|_2 &\le d_1\varepsilon \max\left(\|\zeta_k\widehat{Q}_k\|_2, \|\zeta_k^{-1}\widehat{Q}_k^{-*}\|_2\right)\\
\|F_{k,2}\|_2 &\le d_2\varepsilon\|\widehat{Q}_k\|_2\\
\|F_{k,3}\|_2 &\le d_3\varepsilon\|\widehat{Q}_k^{-*}\|_2\\
\|F_{kb}\|_2 &\le d_b\varepsilon\|\widehat{Q}_k\|_2\\
\|F_{kf}\|_2 &\le d_f\frac{\varepsilon}{2}\max\left(\|\zeta_k\widehat{Q}_k\|_2, \|\zeta_k^{-1}\widehat{Q}_k^{-*}\|_2\right),
\end{aligned}$$

where ε is the machine epsilon and d_1, $d_2 = d_b$, d_3, and d_f are some modest constants that may depend on n, the details of the arithmetic and the inversion algorithm but depend neither on Q_k nor $\widehat{Q}_k$. Each Q_k is a smooth function of Q_{k-1} and each $F_{k,j} = O(\varepsilon)$, so, by induction,

$$\widehat{Q}_k = Q_k + O(\varepsilon). \tag{4.2}$$

Hence, the bounds above may be loosely expressed in terms of Q_k as

$$\|F_{kb}\|_2 \le d_b\varepsilon\|Q_k\|_2 + O(\varepsilon^2) \tag{4.3}$$

$$\begin{aligned}
\|F_{kf}\|_2 &\le d_f\frac{\varepsilon}{2}\max\left(\|\zeta_k Q_k\|_2, \|\zeta_k^{-1}Q_k^{-*}\|_2\right) + O(\varepsilon^2)\\
&\le d_f\varepsilon\|Q_{k+1}\|_2 + O(\varepsilon^2).
\end{aligned} \tag{4.4}$$

Inequality (4.4) is a consequence of (2.2).

We need the following lemma for continuing our analysis.

LEMMA 4.3. Let $A \in \mathbb{C}^{n\times n}$ [illegible] $A = QH$ [illegible] $Q \in \mathbb{C}^{n\times n}$ [illegible] $H \in \mathbb{C}^{n\times n}$ [illegible] $F \in \mathbb{C}^{n\times n}$ [illegible] $\|F\|_2 = 1$ [illegible] $t \ge 0$ [illegible] t [illegible] $A + tF$ [illegible]

$$A + tF = Q\left((I + tE) + O(t^2)\right)\left(H + tG + O(t^2)\right),$$

[illegible] $G \in \mathbb{C}^{n\times n}$ [illegible]

$$F^*A + A^*F = GH + HG, \tag{4.5}$$

[illegible] $E \in \mathbb{C}^{n\times n}$ [illegible]

$$Q^*F - F^*Q = EH + HE. \tag{4.6}$$

[illegible] E [illegible]

$$E = (Q^*F - G)H^{-1}. \tag{4.7}$$

[illegible]. See proof in Appendix A.2. □

At each of the perturbed Newton iteration (4.1) rounding errors are equivalent to perturbing $\widehat{Q}_k$ to $\widehat{Q}_k + F_{kb}$, taking one Newton step (1.1), then perturbing the result by adding F_{kf}. Let $\widehat{Q}_k = W_k\widehat{H}_k$ and $\widehat{Q}_k + F_{kb} = \widetilde{W}_k\widetilde{H}_k$ be the polar decompositions of $\widehat{Q}_k$ and $\widehat{Q}_k + F_{kb}$, respectively. By Lemma 4.3,

$$\widetilde{W}_k = W_k(I + E_{kb}) + O(\varepsilon^2), \tag{4.8}$$

where E_{kb} satisfies

$$E_{kb}\widehat{H}_k + \widehat{H}_k E_{kb} = W_k^* F_{kb} - F_{kb}^* W_k + O(\varepsilon^2). \tag{4.9}$$

From (4.1),

$$\frac{\zeta_k}{2}(\widehat{Q}_k + F_{kb}) + \frac{1}{2\zeta_k}(\widehat{Q}_k + F_{kb})^{-*} = \widehat{Q}_{k+1} - F_{kf}.$$

Since the unitary factor in the polar decomposition of the left-hand side matrix is $\widetilde{W}_k$, applying Lemma 4.3 to $\widehat{Q}_{k+1}$, we have

$$\widetilde{W}_k = W_{k+1}(I - E_{kf}) + O(\varepsilon^2),$$

or equivalently,

$$W_{k+1} = \widetilde{W}_k(I + E_{kf}) + O(\varepsilon^2), \tag{4.10}$$

where E_{kf} satisfies

$$E_{kf}\widehat{H}_{k+1} + \widehat{H}_{k+1}E_{kf} = W_{k+1}^* F_{kf} - F_{kf}^* W_{k+1} + O(\varepsilon^2). \tag{4.11}$$

Since Q_k has the polar decomposition $Q_k = QH_k$ and $\widehat{Q}_k$ satisfies (4.2), by Lemma 4.3 one also has $\widehat{H}_k = H_k + O(\varepsilon)$, $W_k = Q + O(\varepsilon)$. Based on these first order error results, (4.9) and (4.11) can be expressed as

$$E_{kb}H_k + H_k E_{kb} = Q^* F_{kb} - F_{kb}^* Q + O(\varepsilon^2), \tag{4.12}$$

$$E_{kf}H_{k+1} + H_{k+1}E_{kf} = Q^* F_{kf} - F_{kf}^* Q + O(\varepsilon^2). \tag{4.13}$$

Combining (4.8) and (4.10), one has

$$W_{k+1} = W_k(I + E_{kb} + E_{kf}) + O(\varepsilon^2).$$

It follows by induction (with $W_0 = Q$) that

$$W_j = Q(I + E_j) + O(\varepsilon^2), \quad j > 0,$$

where

$$E_j = \sum_{k=0}^{j-1}(E_{kb} + E_{kf}). \tag{4.14}$$

Suppose that Algorithm 3.1 applied to a nonsingular matrix $A \in \mathbb{C}^{n\times n}$, with polar decomposition $A = QH$ completes after p iterations in Step 1. We obtain $\widehat{Q}_p$ that satisfies $\|\widehat{Q}_p - \widehat{Q}_p^{-*}\|_2 \le \|\widehat{Q}_p - \widehat{Q}_p^{-*}\|_F < \delta$. With the polar decomposition $\widehat{Q}_p = W_p\widehat{H}_p$ and (2.2), we have (see the proof in Appendix A.3)

$$\frac{1}{2}(\widehat{Q}_p + \widehat{Q}_p^{-*}) = W_p + \Delta\widehat{Q}_p, \tag{4.15}$$

where

$$\|\Delta\widehat{Q}_p\|_2 < \delta^2/8.$$

Suppose δ is small. Step 2 of the algorithm produces approximate polar factors

$$\widehat{Q} = \frac{1}{2}(\widehat{Q}_p + \widehat{Q}_p^{-*}) + F, \quad \widehat{H} = \frac{1}{2}\left(\widehat{Q}^* A + A^* \widehat{Q} + K\right),$$

where F accounts for rounding error forming $\widehat{Q}$ from $\widehat{Q}_p$ and K for rounding error forming $\widehat{H}$ from A and $\widehat{Q}$ obeying

$$\|F\| \le d_F \varepsilon, \qquad \|K\| \le d_K \varepsilon \|A\|_2,$$

with modest constants d_F and d_K. So we have

$$\|\widehat{Q} - W_p\|_2 \le d_F \varepsilon + \delta^2/8. \tag{4.16}$$

Since $W_p = Q(I+E)$ with $E := E_p$ defined in (4.14), by (4.15),

$$\widehat{Q} = Q(I+E) + \Delta\widehat{Q}_p + F = Q(I+E+L), \tag{4.17}$$

where $L = Q^*(\Delta\widehat{Q}_p + F)$ satisfies

$$\|L\|_2 \le d_L \max(\varepsilon, \delta^2)$$

for some modest constant d_L, which combines the rounding error F and the effect of stopping criterion. Both d_L and d_K may depend on n and the details of the finite precision arithmetic and computational algorithm, but not on A, $\widehat{Q}$, or $\widehat{H}$. With (4.17) and the fact that E is skew-Hermitian,

$$\begin{aligned}
\widehat{H} &= \frac{1}{2}\left((I+E+L)^* Q^* A + A^* Q (I+E+L) + K\right) \\
&= \frac{1}{2}\left(1 - E + L^*\right) H + H(I+E+L) + K) \\
&= \frac{1}{2}(2H - EH + HE + L^* H + HL + K). \qquad (4.18)
\end{aligned}$$

So by (4.17) and (4.18), to the first order,

$$\begin{aligned}
\widehat{Q}\widehat{H} - A &= \frac{1}{2} Q(I+E+L)(2H - EH + HE + L^*H + HL + K) - A \\
&= \frac{1}{2} Q\left(2H - EH + HE + L^*H + HL + 2EH + 2LH + K\right) - A \\
&\quad + O(\|L\|_2^2) + O(\varepsilon\|L\|_2) + O(\varepsilon^2) \\
&= \frac{1}{2} Q\left(EH + HE + L^*H + HL + 2LH + K\right) + O(\max(\varepsilon^2, \varepsilon\delta^2, \delta^4)).
\end{aligned}$$

From (4.14) this expression can be written

$$\begin{aligned}
\widehat{Q}\widehat{H} - A &= \frac{1}{2} Q \sum_{k=0}^{p-1} (E_{kb} H + H E_{kb} + E_{kf} H + H E_{kf}) \\
&\quad + \frac{1}{2} Q(L^* H + HL + 2LH + K) + O(\max(\varepsilon^2, \varepsilon\delta^2, \delta^4)). \qquad (4.19)
\end{aligned}$$

Note that so far the suboptimal scale factors have not played a role. In order to continue the analysis, we need the following lemma which involves the suboptimal scaling.

LEMMA 4.4. Let $A \in \mathbb{C}^{n\times n}$ have the singular value decomposition $U\Sigma V^*$, $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$, and let (2.1) and (1.6) hold. Let $\Sigma_0 = \Sigma$ and $\sigma_j^{(k)}$ be the jth diagonal entry of Σ_k. Let $\sigma_{\max}^{(k)} = \max_{1\leq j\leq n} \sigma_j^{(k)}$. Let a, b satisfy $0 < a \leq \sigma_n$ and $b \geq \sigma_1$. Then for all $k \geq 0$ and $1 \leq i, j \leq n$,

$$\frac{\sigma_{\max}^{(k)}}{\sigma_i^{(k)} + \sigma_j^{(k)}} \leq \frac{b}{\sigma_i + \sigma_j}.$$

Proof. See Appendix A.4. ∎

Recall that E_{kb}, E_{kf} satisfy (4.12) and (4.13), respectively. Let

$$H_k = V\Sigma_k V^*$$

be an eigen-decomposition, where V is unitary and Σ_k is diagonal obeying (2.1). (Recall that throughout the algorithm the singular vectors of the Q_k's are the singular vectors of A. In particular, for all k, the unitary matrix of the right singular vectors of A is also a unitary matrix of the eigenvectors of H_k.) In this notation, (4.12) and (4.13) can be written

$$\widetilde{E}_{kb}\Sigma_k + \Sigma_k\widetilde{E}_{kb} = \widetilde{F}_{kb} - \widetilde{F}_{kb}^* + O(\varepsilon^2),$$
$$\widetilde{E}_{kf}\Sigma_{k+1} + \Sigma_{k+1}\widetilde{E}_{kf} = \widetilde{F}_{kf} - \widetilde{F}_{kf}^* + O(\varepsilon^2),$$

where

$$\widetilde{E}_{kb} = V^*E_{kb}V,\ \widetilde{E}_{kf} = V^*E_{kf}V,\ \widetilde{F}_{kb} = V^*Q^*F_{kb}V,\ \widetilde{F}_{kf} = V^*Q^*F_{kf}V. \tag{4.20}$$

So, the (i,j)th entries of $\widetilde{E}_{kb}$ and $\widetilde{E}_{kf}$ are

$$\widetilde{e}_{ij,kb} = \frac{\widetilde{f}_{ij,kb} - \overline{\widetilde{f}_{ji,kb}}}{\sigma_i^{(k)} + \sigma_j^{(k)}} + O(\varepsilon^2), \qquad \widetilde{e}_{ij,kf} = \frac{\widetilde{f}_{ij,kf} - \overline{\widetilde{f}_{ji,kf}}}{\sigma_i^{(k+1)} + \sigma_j^{(k+1)}} + O(\varepsilon^2). \tag{4.21}$$

Note that $\|\widetilde{E}_{kj}\|_2 = \|E_{kj}\|_2$ and $\|\widetilde{F}_{kj}\|_2 = \|F_{kj}\|_2$ for $j = b, f$, because V and Q are unitary.

Multiplying (4.19) on the left by V^* and on the right by V gives

$$\begin{aligned} V^*(\widehat{Q}\widehat{H} - A)V = {} & \frac{1}{2}V^*QV\sum_{k=0}^{p-1}\left(\widetilde{E}_{kb}\Sigma + \Sigma\widetilde{E}_{kb} + \widetilde{E}_{kf}\Sigma + \Sigma\widetilde{E}_{kf}\right) \\ & + \frac{1}{2}V^*\left(L^*H + HL + 2LH + K\right)V + O(\max(\varepsilon^2, \varepsilon\delta^2, \delta^4)), \end{aligned}$$

where $\widetilde{E}_{kb}$ and $\widetilde{E}_{kf}$ are given by (4.20). From (4.21), the (i,j)th entry of the sum $\sum_{k=0}^{p-1}(\widetilde{E}_{kb}\Sigma + \Sigma\widetilde{E}_{kb} + \widetilde{E}_{kf}\Sigma + \Sigma\widetilde{E}_{kf})$ is

$$\begin{aligned} & \sum_{k=0}^{p-1}(\widetilde{e}_{ij,kb} + \widetilde{e}_{ij,kf})(\sigma_i + \sigma_j) \\ & = \sum_{k=0}^{p-1}\left(\frac{\widetilde{f}_{ij,kb} - \overline{\widetilde{f}_{ji,kb}}}{\sigma_i^{(k)} + \sigma_j^{(k)}} + \frac{\widetilde{f}_{ij,kf} - \overline{\widetilde{f}_{ji,kf}}}{\sigma_i^{(k+1)} + \sigma_j^{(k+1)}}\right)(\sigma_i + \sigma_j) + O(\varepsilon^2) \\ & = \sum_{k=0}^{p-1}\left(\frac{\widetilde{f}_{ij,kb} - \overline{\widetilde{f}_{ji,kb}}}{\sigma_{\max}^{(k)}}\,\frac{(\sigma_i + \sigma_j)\sigma_{\max}^{(k)}}{\sigma_i^{(k)} + \sigma_j^{(k)}} + \frac{\widetilde{f}_{ij,kf} - \overline{\widetilde{f}_{ji,kf}}}{\sigma_{\max}^{(k+1)}}\,\frac{(\sigma_i + \sigma_j)\sigma_{\max}^{(k+1)}}{\sigma_i^{(k+1)} + \sigma_j^{(k+1)}}\right) \\ & \quad + O(\varepsilon^2). \end{aligned}$$

Inequalities (4.3) and (4.4) and Lemma 4.4 imply

$$\begin{aligned}&\left|\sum_{k=0}^{p-1}(\widetilde{e}_{ij,kb}+\widetilde{e}_{ij,kf})(\sigma_i+\sigma_j)\right|\\&\le 2\varepsilon\sum_{k=0}^{p-1}\left(d_b\frac{(\sigma_i+\sigma_j)\sigma_{\max}^{(k)}}{\sigma_i^{(k)}+\sigma_j^{(k)}}+d_f\frac{(\sigma_i+\sigma_j)\sigma_{\max}^{(k+1)}}{\sigma_i^{(k+1)}+\sigma_j^{(k+1)}}\right)+O(\varepsilon^2)\\&\le 2p\varepsilon(d_b+d_f)b+O(\varepsilon^2).\end{aligned}$$

Hence, the residual is bounded as

$$\begin{aligned}\|\widehat{Q}\widehat{H}-A\|_2&\le\|Q\sum_{k=0}^{p-1}\left(E_{kb}H+HE_{kb}+E_{kf}H+HE_{kf}\right)/2\|_2\\&\quad+\|Q(L^*H+HL+2LH+K)/2\|_2+O(\max(\varepsilon^2,\varepsilon\delta^2,\delta^4))\\&\le np\varepsilon(d_b+d_f)b+(2d_L+d_K/2)\max(\varepsilon,\delta^2)\|A\|_2+O(\max(\varepsilon^2,\varepsilon\delta^2,\delta^4)).\end{aligned}$$

In the same way, from (4.18) we can obtain

$$\begin{aligned}\|\widehat{H}-H\|_2&\le\|(-EH+HE)/2\|_2+\|(L^*H+HL+K)/2\|_2+O(\delta^4)\\&\le np\varepsilon(d_b+d_f)b+(d_L+d_K/2)\max(\varepsilon,\delta^2)\|A\|_2+O(\max(\varepsilon^2,\varepsilon\delta^2,\delta^4)).\end{aligned}$$

By applying Lemma 4.4 to (4.21) to estimate $\|E\|_2$, then from (4.17) we can derive

$$\|\widehat{Q}-Q\|_2\le\|QE\|_2+\|QL\|_2\le\theta np\varepsilon(d_b+d_f)b+d_L\max(\varepsilon,\delta^2),$$

where

$$\theta=\begin{cases}\frac{1}{\sigma_n} & A\text{ is complex,}\\ \frac{2}{\sigma_{n-1}+\sigma_n} & A\text{ is real.}\end{cases}\tag{4.22}$$

(The formula in the real case is based on the fact that $\widetilde{e}_{ii,kb}=\widetilde{e}_{ii,kf}=0$ from (4.21).)

We present the above error analysis results as well as (4.16) in the following theorem.

THEOREM 4.5. [illegible] $A\in\mathbb{C}^{n\times n}$ [illegible] 4.2 [illegible] $A=QH$ [illegible] $\widehat{Q}$ [illegible] $\widehat{H}$ [illegible] 3.1 [illegible] p [illegible] 4.1 [illegible]

$$\begin{aligned}\|\widehat{Q}\widehat{H}-A\|_2&\le np\varepsilon(d_b+d_f)b+(2d_L+d_K/2)\max(\varepsilon,\delta^2)\|A\|_2+O(\max(\varepsilon^2,\varepsilon\delta^2,\delta^4))\\\|\widehat{H}-H\|_2&\le np\varepsilon(d_b+d_f)b+(d_L+d_K/2)\max(\varepsilon,\delta^2)\|A\|_2+O(\max(\varepsilon^2,\varepsilon\delta^2,\delta^4))\\\|\widehat{Q}-Q\|_2&\le\theta np\varepsilon(d_b+d_f)b+d_L\max(\varepsilon,\delta^2)\\\|\widehat{Q}-W_p\|_2&\le d_F\varepsilon+\delta^2/8,\end{aligned}$$

[illegible] d_b,d_f,d_L,d_K,d_F [illegible] W_p [illegible] θ [illegible] (4.22)

As noted in Table 2.1, in most practical situations $p\le 9$. Therefore, if b is not too much greater than $\|A\|_2$ and the algorithm uses a stopping criterion δ not too much greater than $\sqrt{\varepsilon}$, then the algorithm is backward stable.

COROLLARY 4.6. , $b = \|A\|_2$, , $\delta = n^{\frac{1}{4}}\sqrt{\varepsilon}$, , ,

$$\|\widehat{Q}\widehat{H} - A\|_2 \le (np(d_b + d_f) + \sqrt{n}(2d_L + d_K/2))\varepsilon\|A\|_2 + O(\varepsilon^2)$$

$$\|\widehat{H} - H\|_2 \le (np(d_b + d_f) + \sqrt{n}(d_L + d_K/2))\varepsilon\|A\|_2 + O(\varepsilon^2)$$

$$\|\widehat{Q} - Q\|_2 \le np(d_b + d_f)\varepsilon(\theta\|A\|_2) + \sqrt{n}d_L\varepsilon$$

$$\|\widehat{Q} - W_p\|_2 \le (d_F + \sqrt{n}/8)\varepsilon.$$

Note that the error bounds for $\|\widehat{H} - H\|_2$ and $\|\widehat{Q} - Q\|_2$ coincide with the perturbation results; see, for example, [8, 21, 20] and [11, section 8.2]. The quantity $\theta\|A\|_2$ serves as the condition number for the perturbation of Q.

, , , 1. The same procedure can be used to give an error analysis for Newton's method with other scalings. Note that the backward stability depends on whether

$$\frac{(\sigma_i + \sigma_j)\sigma_{\max}^{(k)}}{\sigma_i^{(k)} + \sigma_j^{(k)}} = O(\|A\|_2), \tag{4.23}$$

which depends on scaling factors. From Remark 2 in Appendix A.4, (4.23) holds for the optimal scaling (1.2). Lemma A.3 in Appendix A.5 shows that (4.23) also holds for the $(1, \infty)$-scaling (1.3) and the scaling (1.4). Therefore, Newton's method with these three scalings is also backward stable under the same conditions of Theorem 4.5 and an appropriate stopping criterion, when the number of iterations is not too large. (See Remark 3 in Appendix A.5.)

We also observed that the numerical stability doesn't necessarily depend on how fast the method converges. In fact, one can show that when $\|A\|_2 \ge \|A^{-1}\|_2$, Newton's method without scaling ($a = b = 1$) computes a polar decomposition satisfying the same error bounds given in Theorem 4.5.

5. Numerical examples. We did some numerical experiments with Newton's method (1.1) with scaling (1.6) using $a = \|A^{-1}\|_F^{-1}$, $b = \|A\|_F$, and also with the $(1, \infty)$-scaling (1.3). The main purpose is to test the numerical stability results and convergence rate and to compare the suboptimal scaling and $(1, \infty)$-scaling. For this reason we used the bidiagonal reduction matrix inversion method Algorithm BR for computing matrix inverses.

All numerical experiments were done on a Dell personal computer with a Pentium-IV processor, in MATLAB version 7.2 with machine epsilon $\varepsilon \approx 2.22 \times 10^{-16}$.

In the numerical experiments we use stopping criterion $\delta = n^{\frac{1}{4}}\sqrt{\varepsilon}$, where n is the size of matrices. For Newton's method with the $(1, \infty)$-scaling, the scale factor is shifted to 1 when $\|X_{k+1} - X_k\|_F / \|X_{k+1}\|_F < 10^{-2}$. Based on the results in Corollary 4.6, if $\widehat{Q}$ and $\widehat{H}$ are the computed unitary and Hermitian polar factors produced by Algorithm 3.1, then we expect to observe that $\|A - \widehat{Q}\widehat{H}\|_2/\|A\|_2$, $\|H - \widehat{H}\|_2/\|H\|_2$, and $\|Q - \widehat{Q}\|_2/(\theta\|A\|_2)$ are not much larger than ε.

In the tables we will use the following notations:

$$e_Q = \frac{\|Q - \widehat{Q}\|_2}{\theta\|A\|_2}, \quad e_H = \frac{\|H - \widehat{H}\|_2}{\|H\|_2}, \quad res = \frac{\|A - \widehat{Q}\widehat{H}\|_2}{\|A\|_2}, \quad ror = \|\widehat{Q}^*\widehat{Q} - I\|_2,$$

where the "exact" factors Q and H for the matrices in the first example are obtained from the SVD of A using MATLAB's variable precision arithmetic `vpa` with 24 signi-

TABLE 5.1
The extreme values of errors, residuals, and iteration counts from Example 1.

kappa		10^2		10^8		10^{15}	
		Min	Max	Min	Max	Min	Max
p	SUB	6	6	8	8	8	9
	HSF	6	7	8	8	8	9
e_Q	SUB	2.6e−17	7.2e−17	7.4e−19	1.7e−17	6.9e−19	2.3e−17
	HSF	2.2e−17	7.7e−17	7.4e−19	1.7e−17	6.9e−19	2.3e−17
e_H	SUB	2.2e−16	4.1e−16	2.5e−16	4.1e−16	1.9e−16	4.4e−16
	HSF	1.7e−16	3.9e−16	1.9e−16	3.9e−16	2.2e−16	4.0e−16
res	SUB	4.1e−16	8.9e−16	4.0e−16	7.5e−16	3.5e−16	6.3e−16
	HSF	3.5e−16	8.2e−16	2.7e−16	5.9e−16	3.9e−16	7.2e−16
ror	SUB	7.9e−16	1.1e−15	8.0e−16	1.1e−15	7.8e−16	1.3e−15
	HSF	7.0e−16	1.2e−15	7.9e−16	1.2e−15	8.2e−16	1.1e−15

TABLE 5.2
Errors, residuals, and iteration counts for Hilbert matrices from Example 2.

n		6	8	10	12	14
p	SUB	8	8	9	9	9
	HSF	7	8	8	9	9
e_Q	SUB	1.2e−18	1.6e−19	2.7e−19	4.1e−19	2.0e−17
	HSF	1.2e−18	1.6e−19	2.7e−19	4.1e−19	2.0e−17
e_H	SUB	2.3e−16	1.9e−16	8.7e−17	1.4e−16	2.3e−16
	HSF	1.9e−16	1.4e−16	8.7e−17	1.6e−16	2.7e−16
res	SUB	2.6e−16	2.4e−16	1.8e−16	3.0e−16	3.8e−16
	HSF	2.5e−16	2.5e−16	1.8e−16	3.3e−16	6.3e−16
ror	SUB	2.6e−16	3.9e−16	6.2e−16	6.3e−16	6.5e−16
	HSF	2.8e−16	5.3e−16	6.8e−16	8.5e−16	1.0e−15

ficant decimal digits. p is the number of iterations, and n is the dimension of matrices. The symbol "HSF" refers to Newton's method (1.1) with Higham's $(1, \infty)$-scaling. The symbol "SUB" refers to (1.1) with scaling (1.6) using the Frobenius norms for the initial interval, i.e., $a = \|A^{-1}\|_F^{-1}$ and $b = \|A\|_F$.

1. Three groups of twenty real matrices were constructed with dimension 20 by using MATLAB's `gallery('randsvd',20,kappa,5)`, with `kappa` equal to $10^2, 10^8, 10^{15}$, respectively. The singular values of the generated matrices are random values with uniformly distributed logarithm. For each group the ranges of the condition numbers $\kappa_2(A)$ and $\theta\|A\|_2$ are listed below:

`kappa` $= 10^2 : \kappa_2(A) \in [37.7,\ 90.6]$, $\theta\|A\|_2 \in [33.5,\ 83.7]$,
`kappa` $= 10^8 : \kappa_2(A) \in [1.13,\ 6.75] \times 10^7$, $\theta\|A\|_2 \in [0.2,\ 5.44] \times 10^7$,
`kappa` $= 10^{15} : \kappa_2(A) \in [6.35 \times 10^{10},\ 7.53 \times 10^{14}]$, $\theta\|A\|_2 \in [1.78 \times 10^{10},\ 5.22 \times 10^{14}]$.

The test results are summarized in Table 5.1, where, for each group, the minimum and maximum values of the errors, residuals, and numbers of iterations are listed.

2. In this example the test matrices are the Hilbert matrices, which are $n \times n$ matrices with entries $a_{ij} = 1/(i + j - 1)$. The example uses dimensions $n = 6$, $n = 8$, $n = 10$, $n = 12$, and $n = 14$. For every Hilbert matrix, the polar decomposition is $A_n = I_n A_n$.

The condition number $\kappa_2(A_n)$ ranges from 1.5×10^7 to 5.1×10^{17}, and $\theta\|A_n\|_2$ ranges from 2.6×10^5 to 1.0×10^{17}. The test results are reported in Table 5.2.

Newton's method with the suboptimal scaling performed well in both examples calculating the polar factors to nearly full precision. As predicted it takes at most 9 iterations. Newton's method with the $(1, \infty)$-scaling performs equally well.

6. Conclusion. The suboptimal scaling scheme (1.6) is essentially costless and simplifies the algorithm of Newton's iteration (1.1) for computing polar factors. In a typical floating point system, with this scaling scheme, for matrices with $\kappa_2(A) < 10^{16}$, no more than 9 iterations are needed for convergence to the unitary polar factor with a convergence tolerance roughly equal to the machine epsilon. By employing the bidiagonal factorization for matrix inversion, (1.1) with (1.6) forms a provably backward stable algorithm. Newton's method with $(1, \infty)$-scaling and scaling (1.4) is also proved to be backward stable, provided the number of iterations is not too large.

Appendix A.

A.1. Bidiagonal reduction-based matrix inversion algorithm.

ALGORITHM BR.
Input: Nonsingular matrix $A \in \mathbb{C}^{n\times n}$
Output: $G = A^{-1}$

Step 1: Compute $A = UBV^*$, with U, V unitary and B upper bidiagonal.
Step 2: Solve $BY = U^*$ for Y by back substitution.
Step 3: Compute $G = VY$.

In Step 1 one may use the Householder reflectors to perform the reduction. The reduction needs $\frac{8}{3}n^3$ flops and computing U needs $\frac{4}{3}n^3$ flops. The matrix V is stored in factorized form. The cost for solving the matrix equation is $O(n^2)$ flops. With the factorized form of V, it needs $2n^3$ flops to compute G. So the total cost is $6n^3$ flops.

In order to show that a matrix inverse computed by Algorithm BR follows Assumption 4.1, we need the following lemma.

LEMMA A.1. [illegible] $Bx = z$ [illegible] $z \in \mathbb{C}^n$ [illegible] B [illegible]

$$B = \begin{bmatrix} \alpha_1 & \beta_1 & & 0 \\ & & & \\ & & & \beta_{n-1} \\ 0 & & & \alpha_n \end{bmatrix}.$$

[illegible] $\hat{x}$ [illegible] $\hat{x}$ [illegible]

$$\hat{x} = B^{-1}(z + \delta z) + \delta x,$$

[illegible]

$$|\delta z| \le 3n\varepsilon|z| + O(\varepsilon^2), \qquad |\delta x| \le 3n\varepsilon|\hat{x}| + O(\varepsilon^2).$$

[illegible]. The components of the computed vector $\hat{x}$ can be formulated as

$$\hat{x}_n = \frac{z_n}{\alpha_n(1+\epsilon_n)}, \qquad \hat{x}_k = \frac{z_k(1+\delta_k) - \beta_k\hat{x}_{k+1}}{\alpha_k(1+\epsilon_k)}, \quad 1 \le k \le n-1,$$

where $|\epsilon_n|, |\delta_k| < \varepsilon$, $|\epsilon_k| < 3\varepsilon$, $k \le n-1$. So we have

$$\begin{aligned} &\alpha_n\hat{x}_n(1+\epsilon_n) = z_n \\ &\alpha_{n-1}\hat{x}_{n-1}(1+\epsilon_{n-1}) + \beta_{n-1}\hat{x}_n = z_{n-1}(1+\delta_{n-1}) \\ &\qquad\vdots \\ &\alpha_1\hat{x}_1(1+\epsilon_1) + \beta_1\hat{x}_2 = z_1(1+\delta_1). \end{aligned} \tag{A.1}$$

Define $\tilde{x} = [\tilde{x}_1, \ldots, \tilde{x}_n]^T$, $\tilde{z} = [\tilde{z}_1, \ldots, \tilde{z}_n]^T$, with

$$\tilde{x}_n = \hat{x}_n(1+\epsilon_n), \quad \tilde{x}_{n-1} = \hat{x}_{n-1}(1+\epsilon_{n-1})(1+\epsilon_n), \quad \ldots, \quad \tilde{x}_1 = \hat{x}_1 \prod_{k=1}^{n}(1+\epsilon_k),$$

$$\tilde{z}_n = z_n, \quad \tilde{z}_{n-1} = z_{n-1}(1+\delta_{n-1})(1+\epsilon_n), \quad \ldots, \quad \tilde{z}_1 = z_1(1+\delta_1)\prod_{k=2}^{n}(1+\epsilon_k).$$

By multiplying $(1+\epsilon_n)$ to the second equation, $(1+\epsilon_n)(1+\epsilon_{n-1})$ to the third equation, and so on in (A.1), we obtain that $\tilde{x}$ and $\tilde{z}$ satisfy

$$B\tilde{x} = \tilde{z}.$$

Let $\delta z = \tilde{z} - z$, and $\delta x = \hat{x} - \tilde{x}$. Then from $\tilde{x} = B^{-1}\tilde{z}$ we have

$$\hat{x} = B^{-1}(z + \delta z) + \delta x.$$

The error bounds for $|\delta x|$ and $|\delta z|$ follow simply from the definitions. □

THEOREM A.2. [illegible] 4.1 [illegible]. We only consider the first order errors.

Let $\hat{U}\hat{B}\hat{V}^*$ be the computed bidiagonal factorization of A. Then $\hat{U} = U + \Delta U_1$, $\hat{V} = V + \Delta V_1$, where U, V are unitary and $\|\Delta U_1\|_2 \le d_1\varepsilon$, $\|\Delta V_1\|_2 \le d_2\varepsilon$, and

$$U\hat{B}V^* = A + E,$$

where $\|E\|_2 \le d_3\varepsilon\|A\|_2$ for some modest constants d_1, d_2, d_3. Let $\hat{Y}$ be the numerical solution of $\hat{B}Y = \hat{U}^*$ computed by back substitution. By Lemma A.1,

$$\hat{Y} = \hat{B}^{-1}(\hat{U}^* + \Delta U_2) + \Delta Y,$$

where

$$\|\Delta U_2\|_2 \le 3n^{\frac{3}{2}}\varepsilon, \qquad \|\Delta Y\|_2 \le 3n^{\frac{3}{2}}\varepsilon\|\hat{Y}\|_2.$$

Let X be the computed matrix product $\hat{V}\hat{Y}$. We have

$$X = \hat{V}\hat{Y} + \Delta X,$$

where $\|\Delta X\|_2 \le d_4\varepsilon\|\hat{Y}\|_2$ for some modest constant d_4. Now

$$\begin{aligned} X &= \hat{V}\hat{B}^{-1}(\hat{U}^* + \Delta U_2) + \hat{V}\Delta Y + \Delta X = \hat{V}\hat{B}^{-1}\hat{U}^* + \hat{V}\hat{B}^{-1}\Delta U_2 + \hat{V}\Delta Y + \Delta X \\ &=: V\hat{B}^{-1}U^* + F = (A+E)^{-1} + F, \end{aligned}$$

where

$$F = \Delta V_1\hat{B}^{-1}U^* + V\hat{B}^{-1}(\Delta U_1)^* + \hat{V}\hat{B}^{-1}\Delta U_2 + \hat{V}\Delta Y + \Delta X.$$

It is easily verified that $\|F\|_F \le (6n^{\frac{3}{2}} + d_1 + d_2 + d_4)\varepsilon\|A^{-1}\|_2$. □

A.2. Proof of Lemma 4.3. Equations (4.5) and (4.7) are established in the proof of Theorem 2.5 in [8]. Here we slightly modify that proof to establish (4.6).

Let $A(t) = A + tF$ have polar decompositions $Q(t)H(t)$. Note that $H(t) = (A(t)^*A(t))^{1/2}$ (positive definite square root) and $Q(t) = A(t)H(t)^{-1}$ are sums, differences, products, quotients, and compositions with C^∞ functions of the entries of $A(t)$, which is trivially a C^∞ function. Hence, $Q(t)$ and $H(t)$ are also C^∞. (Here we use the fact that A is nonsingular to observe that $H(t) = (A(t)^*A(t))^{1/2}$ avoids the singularity of the square root at zero.) Using $\dot{Q}$ and $\dot{H}$ to denote differentiation by t, Taylor's theorem implies that

$$\begin{aligned} Q(t) &= Q(0) + t\dot{Q}(0) + O(t^2) = Q(I + tQ^*\dot{Q}(0)) + O(t^2) \\ H(t) &= H(0) + t\dot{H}(0) + O(t^2). \end{aligned}$$

The proof of Theorem 2.5 in [8] shows $\dot{H}(0) = G = G^*$, with G given by (4.5) and $Q^*(0)\dot{Q}(0) = E = -E^*$, with E given by (4.7).

Differentiate $A + tF = Q(t)H(t)$ to get $F = \dot{Q}H + Q\dot{H}$. Evaluating at $t = 0$, letting $E = Q^*(0)\dot{Q}(0)$, $G = \dot{H}(0)$ gives $Q^*F = EH + G$. Using the facts that E is skew-Hermitian and G is Hermitian while subtracting this equation to its Hermitian transpose gives (4.6). The Lyapunov operator on the right is nonsingular, because A nonsingular implies that the eigenvalues of the Hermitian polar factor H are real and positive. Hence, the solution E is unique.

A.3. Proof for (4.15). Let $\widehat{Q}_p = U_p\Sigma_pV_p^*$ be the SVD. Recall that the singular values satisfy $\sigma_i^{(p)} \geq 1$. Then $\|\widehat{Q}_p - \widehat{Q}_p^{-*}\|_2 < \delta$ implies that

$$\sigma_i^{(p)} - \frac{1}{\sigma_i^{(p)}} < \delta, \quad i = 1, 2, \ldots, n.$$

Because

$$\sigma_i^{(p)} - \frac{1}{\sigma_i^{(p)}} = \frac{(\sigma_i^{(p)} + 1)(\sigma_i^{(p)} - 1)}{\sigma_i^{(p)}},$$

we have

$$\sigma_i^{(p)} - 1 < \frac{\delta\sigma_i^{(p)}}{\sigma_i^{(p)} + 1}.$$

Then

$$\frac{1}{2}\left(\sigma_i^{(p)} + \frac{1}{\sigma_i^{(p)}}\right) - 1 = \frac{(\sigma_i^{(p)} - 1)^2}{2\sigma_i^{(p)}} < \frac{\sigma_i^{(p)}}{2(\sigma_i^{(p)} + 1)^2}\delta^2.$$

Since the function $x/(x+1)^2$ is decreasing when $x \geq 1$, we have $\sigma_i^{(p)}/(\sigma_i^{(p)}+1)^2 \leq 1/4$. Hence

$$\frac{1}{2}\left(\sigma_i^{(p)} + \frac{1}{\sigma_i^{(p)}}\right) - 1 < \delta^2/8$$

and using $W_p = U_pV_p^*$,

$$\begin{aligned} \|\Delta\widehat{Q}_p\|_2 &= \|(\widehat{Q}_p + \widehat{Q}_p^{-*})/2 - W_p\|_2 = \|U_p((\Sigma_p + \Sigma_p^{-1})/2 - I)V_p^*\|_2 \\ &= \max_i \left(\frac{1}{2}\left(\sigma_i^{(p)} + \frac{1}{\sigma_i^{(p)}}\right) - 1\right) < \delta^2/8. \end{aligned}$$

A.4. Proof of Lemma 4.4. By (2.4), $\sigma_{\max}^{(k)} \le b_k$ for $k \ge 0$.
So we only need to show that

$$\frac{b_k}{\sigma_i^{(k)} + \sigma_j^{(k)}} \le \frac{b}{\sigma_i + \sigma_j}, \quad k \ge 0.$$

An easy calculation shows, for all i and j,

$$\sigma_i^{(k)} + \sigma_j^{(k)} = \frac{\zeta_{k-1}}{2} \left(\sigma_i^{(k-1)} + \sigma_j^{(k-1)}\right) \left(1 + \frac{1}{\zeta_{k-1}^2 \sigma_i^{(k-1)} \sigma_j^{(k-1)}}\right).$$

Recall b_k satisfies

$$b_k = \frac{1}{2}\left(\zeta_{k-1} b_{k-1} + \frac{1}{\zeta_{k-1} b_{k-1}}\right) = \frac{\zeta_{k-1}}{2}\, b_{k-1} \left(1 + \frac{1}{\zeta_{k-1}^2 b_{k-1}^2}\right).$$

Because $\sigma_i^{(k-1)} \sigma_j^{(k-1)} \le b_{k-1}^2$,

$$\begin{aligned} \frac{b_k}{\sigma_i^{(k)} + \sigma_j^{(k)}} &= \frac{\frac{\zeta_{k-1}}{2}\, b_{k-1} \left(1 + \frac{1}{\zeta_{k-1}^2 b_{k-1}^2}\right)}{\frac{\zeta_{k-1}}{2} \left(\sigma_i^{(k-1)} + \sigma_j^{(k-1)}\right) \left(1 + \frac{1}{\zeta_{k-1}^2 \sigma_i^{(k-1)} \sigma_j^{(k-1)}}\right)} \\ &\le \frac{b_{k-1}}{\sigma_i^{(k-1)} + \sigma_j^{(k-1)}}. \end{aligned}$$

An easy induction on k now implies that

$$\frac{b_k}{\sigma_i^{(k)} + \sigma_j^{(k)}} \le \frac{b_0}{\sigma_i^{(0)} + \sigma_j^{(0)}} = \frac{b}{\sigma_i + \sigma_j}, \quad k \ge 0.$$

. . 2.

1. The condition $a \le \|A^{-1}\|_2^{-1}$ is essential for proving Lemma 4.4. Without it one may not have the inequality $\sigma_j^{(k)} \le b_k$, and the result cannot be proved.
2. In the case that the optimal scaling is employed, $b_k = \sigma_{\max}^{(k)}$, and one has the same result.

A.5. Relation (4.23) for the scalings (1.3) and (1.4).

LEMMA A.3. . . . . Q_k . . . . . ,
. . . . . ζ_k . . . . . $\zeta_k^{(1,\infty)}$. (1.3) . $\zeta_k^{(F)}$. (1.4) . . . . . Q_k . . . .

$$\frac{\sigma_{\max}^{(k)}}{\sigma_i^{(k)} + \sigma_j^{(k)}} \le \left(\prod_{\ell=0}^{k-1} \psi_\ell\right) \frac{\|A\|_2}{\sigma_i + \sigma_j} \le n^{\frac{k}{2}} \frac{\|A\|_2}{\sigma_i + \sigma_j}, \quad k \ge 0, \quad 1 \le i, j \le n, \tag{A.2}$$

. . . $\psi_\ell = \max\{1, \frac{1}{\zeta_\ell^2 \sigma_{\max}^{(\ell)} \sigma_{\min}^{(\ell)}}\} \le \sqrt{n}$.
. . . . Let

$$w_k = \frac{1}{\zeta_k^2 \sigma_{\min}^{(k)} \sigma_{\max}^{(k)}}, \quad k \ge 0.$$

If

$$\zeta_k = \zeta_k^{(1,\infty)} = \sqrt[4]{\frac{\|Q_k^{-1}\|_1 \|Q_k^{-1}\|_\infty}{\|Q_k\|_1 \|Q_k\|_\infty}},$$

using the inequalities $\|A\|_2 \le \sqrt{\|A\|_1 \|A\|_\infty} \le \sqrt{n}\|A\|_2$, we have ([11, p. 208])

$$\frac{1}{\sqrt[4]{n}}\,\zeta_k \le \frac{1}{\sqrt{\sigma_{\min}^{(k)} \sigma_{\max}^{(k)}}} \le \sqrt[4]{n}\,\zeta_k. \tag{A.3}$$

If

$$\zeta_k = \zeta_k^{(F)} = \sqrt{\frac{\|Q_k^{-1}\|_F}{\|Q_k\|_F}},$$

using the inequalities $\|A\|_2 \le \|A\|_F \le \sqrt{n}\,\|A\|_2$, we also have (A.3).

So in both cases we have

$$\frac{1}{\sqrt{n}} \le w_k \le \sqrt{n}.$$

Construct an interval $[a_k, b_k]$ according to the rule

$$\begin{array}{lll} a_k = \sigma_{\min}^{(k)} w_k, & b_k = \sigma_{\max}^{(k)}, & w_k \le 1, \\ a_k = \sigma_{\min}^{(k)}, & b_k = \sigma_{\max}^{(k)} w_k, & w_k \ge 1. \end{array}$$

Because $a_k \le \sigma_{\min}^{(k)}$ and $b_k \ge \sigma_{\max}^{(k)}$, we have $\sigma_1^{(k)}, \dots, \sigma_n^{(k)} \in [a_k, b_k]$. Also, in both cases we have $(\zeta_k a_k)^{-1} = \zeta_k b_k$. So $\rho(\zeta_k a_k) = \rho(\zeta_k b_k)$ and

$$\sigma_j^{(k+1)} = \rho(\zeta_k \sigma_j^{(k)}) \in \rho(\zeta_k [a_k, b_k]) = [1, \rho(\zeta_k b_k)], \quad j = 1, 2, \dots, n.$$

Since

$$\rho(\zeta_k b_k) = \frac{1}{2}\left(\zeta_k b_k + \frac{1}{\zeta_k b_k}\right) = \frac{\zeta_k b_k}{2}\left(1 + \frac{1}{(\zeta_k b_k)^2}\right),$$

we have

$$\begin{aligned} \frac{\sigma_{\max}^{(k+1)}}{\sigma_i^{(k+1)} + \sigma_j^{(k+1)}} \le \frac{\rho(\zeta_k b_k)}{\sigma_i^{(k+1)} + \sigma_j^{(k+1)}} &= \frac{\frac{\zeta_k b_k}{2}\left(1 + \frac{1}{(\zeta_k b_k)^2}\right)}{\frac{\zeta_k}{2}(\sigma_i^{(k)} + \sigma_j^{(k)})\left(1 + \frac{1}{\zeta_k^2 \sigma_i^{(k)} \sigma_j^{(k)}}\right)} \\ &= \frac{b_k\left(1 + \frac{1}{(\zeta_k b_k)^2}\right)}{(\sigma_i^{(k)} + \sigma_j^{(k)})\left(1 + \frac{1}{\zeta_k^2 \sigma_i^{(k)} \sigma_j^{(k)}}\right)}. \end{aligned}$$

If $w_k \le 1$, then $b_k = \sigma_{\max}^{(k)}$. Because $\sigma_i^{(k)} \sigma_j^{(k)} \le \left(\sigma_{\max}^{(k)}\right)^2 = b_k^2$, we have

$$\frac{\sigma_{\max}^{(k+1)}}{\sigma_i^{(k+1)} + \sigma_j^{(k+1)}} \le \frac{b_k}{\sigma_i^{(k)} + \sigma_j^{(k)}} = \frac{\sigma_{\max}^{(k)}}{\sigma_i^{(k)} + \sigma_j^{(k)}}.$$

If $w_k \geq 1$, then $b_k = \sigma_{\max}^{(k)} w_k$. Because $\sigma_i^{(k)}\sigma_j^{(k)} \leq \left(\sigma_{\max}^{(k)}\right)^2 \leq b_k^2$, we have

$$\frac{\sigma_{\max}^{(k+1)}}{\sigma_i^{(k+1)} + \sigma_j^{(k+1)}} \leq \frac{b_k}{\sigma_i^{(k)} + \sigma_j^{(k)}} = w_k \frac{\sigma_{\max}^{(k)}}{\sigma_i^{(k)} + \sigma_j^{(k)}}.$$

Hence

$$\frac{\sigma_{\max}^{(k+1)}}{\sigma_i^{(k+1)} + \sigma_j^{(k+1)}} \leq \psi_k \frac{\sigma_{\max}^{(k)}}{\sigma_i^{(k)} + \sigma_j^{(k)}}, \quad \psi_k = \max\{1, w_k\} \leq \sqrt{n}.$$

Then the inequalities in (A.2) can be easily derived. □

. . . 3. Suppose that Newton's method with scaling $\zeta_k^{(1,\infty)}$ or $\zeta_k^{(F)}$ terminates after p iterations. We have the same error bounds as in Theorem 4.5 but with a factor $n^{\frac{p}{2}}$ in the first term of the bounds for $\|\widehat{Q}\widehat{H} - A\|_2$, $\|\widehat{H} - H\|_2$, and $\|\widehat{Q} - Q\|_2$. When n and p are both large, this factor is notably large, and one may not be able to use the bounds to claim backward stability. Unfortunately, we are unable to provide an upper bound for p, although it is observed that p is usually moderate in practice ($p \leq 9$ for the $(1,\infty)$-scaling for all examples in section 5).

However, we argue that the factor $n^{\frac{p}{2}}$ is an overestimate. The point is that, when Q_k is getting close to a unitary matrix, $\zeta_k^{(1,\infty)}$ and $\zeta_k^{(F)}$ are getting close to 1. Then w_k as well as ψ_k will be close to 1. So in practice w_k will be around 1 after a couple of iterations.

Acknowledgments. We thank Nick Higham for detailed suggestions and sending part of his unpublished book [11]. We thank Andrzej Kiełbasiński for pointing out a mistake in an earlier version and the referees for their comments and suggestions.

REFERENCES

[1] Z. Bai and J. Demmel, *Design of a parallel nonsymmetric eigenroutine toolbox, Part* I, in Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing, R. F. Sincovec et al., eds., SIAM, Philadelphia, 1993, pp. 391–398. Also available as Computer Science Report CSD-92-718, University of California, Berkeley, CA 1992.

[2] Z. Bai and J. Demmel, *Design of a Parallel Nonsymmetric Eigenroutine Toolbox, Part* II, Technical report Department of Mathematics Research Report 95-11, University of Kentucky, Lexington, KY, 1995.

[3] L. A. Balzer, *Accelerated convergence of the matrix sign function method of solving Lyapunov, Riccati and other matrix equations*, Internat. J. Control, 32 (1980), pp. 1057–1078.

[4] R. Byers, *Solving the algebraic Riccati equation with the matrix sign function*, Linear Algebra Appl., 85 (1987), pp. 267–279.

[5] A. A. Dubrulle, *An optimum iteration for the matrix polar decomposition*, Electron. Trans. Numer. Anal., 8 (1999), pp. 21–25 (electronic).

[6] W. Gander, *Algorithms for the polar decomposition*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 1102–1115.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[8] N. J. Higham, *Computing the polar decomposition—with applications*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1160–1174.

[9] N. J. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.

[10] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[11] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008.

[12] N. J. Higham and P. Papadimitriou, *Parallel Singular Value Decomposition via the Polar Decomposition*, Technical report Numerical Analysis Report 239, Department of Mathematics, University of Manchester, Manchester, England, 1993.

[13] N. J. Higham and P. Papadimitriou, *A parallel algorithm for computing the polar decomposition*, Parallel Comput., 20 (1994), pp. 1161–1173.

[14] N. J. Higham and R. S. Schreiber, *Fast polar decomposition of an arbitrary matrix*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 648–655.

[15] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1990, corrected reprint of the 1985 original.

[16] C. S. Kenney and A. J. Laub, *On scaling Newton's method for polar decomposition and the matrix sign function*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 688–706.

[17] C. S. Kenney and A. J. Laub, *The matrix sign function*, IEEE Trans. Automat. Control, 40 (1995), pp. 1330–1348.

[18] A. Kiełbasiński and K. Ziętak, *Numerical behavior of Higham's scaled method for polar decomposition*, Numer. Algorithms, 32 (2003), pp. 105–140.

[19] A. Kiełbasiński, P. Zieliński, and K. Ziętak, *Higham's scaled method for polar decomposition and numerical matrix-inversion*, Technical report Institute of Mathematics and Computer Science Report I18/2007/P-045, Wroclaw University of Technology, Wroclaw, Poland, 2007.

[20] R.-C. Li, *New perturbation bounds for the unitary polar factor*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 327–332.

[21] R. Mathias, *Perturbation bounds for the polar decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 588–597.

[22] J. D. Roberts, *Linear model reduction and solution of the algebraic Riccati equation by use of the sign function*, Internat. J. Control, 32 (1980), pp. 677–687. (Reprint of Technical report TR-13, CUED/B-Control, Cambridge University, Engineering Department, Cambridge, 1971).

[23] P. Zieliński and K. Ziętak, *Polar decomposition—properties, applications and algorithms*, Appl. Math., An. Polish Math. Soc., 38 (1995), pp. 23–49.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 844–881

RELATIVE-ERROR *CUR* MATRIX DECOMPOSITIONS*

PETROS DRINEAS†, MICHAEL W. MAHONEY‡, AND S. MUTHUKRISHNAN§

Abstract. Many data analysis applications deal with large matrices and involve approximating the matrix using a small number of "components." Typically, these components are linear combinations of the rows and columns of the matrix, and are thus difficult to interpret in terms of the original features of the input data. In this paper, we propose and study matrix approximations that are explicitly expressed in terms of a small number of columns and/or rows of the data matrix, and thereby more amenable to interpretation in terms of the original data. Our main algorithmic results are two randomized algorithms which take as input an $m \times n$ matrix A and a rank parameter k. In our first algorithm, C is chosen, and we let $A' = CC^+A$, where C^+ is the Moore–Penrose generalized inverse of C. In our second algorithm C, U, R are chosen, and we let $A' = CUR$. (C and R are matrices that consist of actual columns and rows, respectively, of A, and U is a generalized inverse of their intersection.) For each algorithm, we show that with probability at least $1 - \delta$, $\|A - A'\|_F \leq (1 + \epsilon)\|A - A_k\|_F$, where A_k is the "best" rank-k approximation provided by truncating the SVD of A, and where $\|X\|_F$ is the Frobenius norm of the matrix X. The number of columns of C and rows of R is a low-degree polynomial in k, $1/\epsilon$, and $\log(1/\delta)$. Both the Numerical Linear Algebra community and the Theoretical Computer Science community have studied variants of these matrix decompositions over the last ten years. However, our two algorithms are the first polynomial time algorithms for such low-rank matrix approximations that come with relative-error guarantees; previously, in some cases, it was not even known whether such matrix decompositions exist. Both of our algorithms are simple and they take time of the order needed to approximately compute the top k singular vectors of A. The technical crux of our analysis is a novel, intuitive sampling method we introduce in this paper called "subspace sampling." In subspace sampling, the sampling probabilities depend on the Euclidean norms of the rows of the top singular vectors. This allows us to obtain provable relative-error guarantees by deconvoluting "subspace" information and "size-of-A" information in the input matrix. This technique is likely to be useful for other matrix approximation and data analysis problems.

Key words. *CUR* matrix decomposition, random sampling algorithms, data analysis, approximate least squares

AMS subject classification. 68W20

DOI. 10.1137/07070471X

1. Introduction.

Large $m \times n$ matrices are common in applications since the data often consist of m objects, each of which is described by n features. Examples of object–feature pairs include: documents and words contained in those documents; genomes and environmental conditions under which gene responses are measured; stocks and their associated temporal resolution; hyperspectral images and frequency resolution; and web groups and individual users. In each of these application areas, practitioners spend vast amounts of time analyzing the data in order to understand, interpret, and ultimately use this data for some application-specific task.

*Received by the editors October 8, 2007; accepted for publication (in revised form) by J. G. Nagy April 14, 2008; published electronically September 17, 2008. A preliminary version of this paper appeared in manuscript and technical report format as "Polynomial Time Algorithm for Column-Row-Based Relative-Error Low-Rank Matrix Approximation" [27, 28]. Preliminary versions of parts of this paper have also appeared as conference proceedings [29, 30, 31].

http://www.siam.org/journals/simax/30-2/70471.html

†Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180 (drinep@cs.rpi.edu).

‡Yahoo! Research, Sunnyvale, CA 94089 (mahoney@yahoo-inc.com). Part of this work was performed while at the Department of Mathematics, Yale University, New Haven, CT 06520.

§Google, Inc., New York, NY 10011 (muthu@google.com). Part of this work was performed while at the Department of Computer Science, Rutgers University, New Brunswick, NJ 08854.

Say that A is the $m \times n$ data matrix. In many cases, an important step in data analysis is to construct a compressed representation of A that may be easier to analyze and interpret. The most common such representation is obtained by truncating the SVD at some number $k \ll \min\{m, n\}$ terms, in large part because this provides the "best" rank-k approximation to A when measured with respect to any unitarily invariant matrix norm. Unfortunately, the basis vectors (the so-called eigencolumns and eigenrows) provided by this approximation (and with respect to which every column and row of the original data matrix is expressed) are notoriously difficult to interpret in terms of the underlying data and processes generating that data. For example, the vector $[(1/2)\ \text{age} - (1/\sqrt{2})\ \text{height} + (1/2)\ \text{income}]$, being one of the significant uncorrelated "factors" from a dataset of people's features, is not particularly informative. It would be highly preferable to have a low-rank approximation that is nearly as good as that provided by the SVD but that is expressed in terms of a small number of [illegible] and/or [illegible] of a matrix, rather than linear combinations of those columns and rows.

The main contribution of this paper is to provide such decompositions. In particular, we provide what we call a relative-error [illegible]: given an $m \times n$ matrix A, we decompose it as a product of three matrices, C, U, and R, where C consists of a small number of actual columns of A, R consists of a small number of actual rows of A, and U is a small carefully constructed matrix that guarantees that the product CUR is "close" to A. In fact, CUR will be nearly as good as the best low-rank approximation to A that is traditionally used and that is obtained by truncating the SVD. Hence, the columns of A that are included in C, as well as the rows of A that are included in R, can be used in place of the eigencolumns and eigenrows, with the added benefit of improved interpretability in terms of the original data.

Before describing applications of our main results in the next subsection, we would like to emphasize that two research communities, the Numerical Linear Algebra (NLA) community and the Theoretical Computer Science (TCS) community, have provided significant practical and theoretical motivation for studying variants of these matrix decompositions over the last ten years. In section 3, we provide a detailed treatment of relevant prior work in both the NLA and the TCS literature. The two algorithms presented in this paper are the first polynomial time algorithms for such low-rank matrix approximations that come with relative-error guarantees; previously, in some cases, it was not even known whether such matrix decompositions exist.

1.1. Applications. As an example of this preference for having the data matrix expressed in terms of a small number of actual columns and rows of the original matrix, as opposed to a small number of eigencolumns and eigenrows, consider recent data analysis work in DNA microarray and DNA Single Nucleotide Polymorphism (SNP) analysis [44, 47, 52]. DNA SNP data are often modeled as an $m \times n$ matrix A, where m is the number of individuals in the study, n is the number of SNPs being analyzed, and A_{ij} is an encoding of the jth SNP value for the ith individual. Similarly, for DNA microarray data, m is the number of genes under consideration, n is the number of arrays or environmental conditions, and A_{ij} is the absolute or relative expression level of the ith gene in the jth environmental condition. Biologists typically have an understanding of a single gene that they fail to have about a linear combination of 6000 genes (and also similarly for SNPs, individuals, and arrays); thus, recent work in genetics on DNA microarray and DNA SNP data has focused on heuristics to extract [illegible] genes, environmental conditions, individuals, and SNPs from the eigengenes, eigenconditions, eigenpeople, and eigenSNPs computed from the original

data matrices [44, 47].[1] Our , matrix decomposition is a direct formulation of this problem: determine a small number of actual SNPs that serve as a basis with which to express the remaining SNPs, and a small number of individuals to serve as a basis with which to express the remaining individuals. In fact, motivated in part by this, we have successfully applied a variant of the , matrix decomposition presented in this paper to intra- and interpopulation genotype reconstruction from tagging SNPs in DNA SNP data from a geographically diverse set of populations [52]. In addition, we have applied a different variant of our , matrix decomposition to hyperspectrally resolved medical imaging data [48]. In this application, a column corresponds to an image at a single physical frequency and a row corresponds to a single spectrally resolved pixel, and we have shown that data reconstruction and classification tasks can be performed with little loss in quality even after substantial data compression [48].

A quite different motivation for low-rank matrix approximations expressed in terms of a small number of columns and/or rows of the original matrix is to decompose efficiently large low-rank matrices that possess additional structure such as sparsity or nonnegativity. This often arises in the analysis of, e.g., large term-document matrices [58, 59, 8]. Another motivation comes from statistical learning theory, where the data need not even be elements in a vector space, and thus expressing the Gram matrix in terms of a small number of actual data points is of interest [64, 63, 24, 25]. This procedure has been shown empirically to perform well for approximate Gaussian process classification and regression [64], to approximate the solution of spectral partitioning for image and video segmentation [32], and to extend the eigenfunctions of a data-dependent kernel to new data points [7, 45]. Yet another motivation is provided by integral equation applications [40, 39, 38], where large coefficient matrices arise that have blocks corresponding to regions where the kernel is smooth and that are thus well-approximated by low-rank matrices. In these applications, partial SVD algorithms can be expensive, and a description in terms of actual columns and/or rows is of interest [39, 38]. A final motivation for studying matrix decompositions of this form is to obtain low-rank matrix approximations to extremely large matrices where a computation of the SVD is too expensive [33, 34, 21, 22, 23].

1.2. Our main results. Our main algorithmic results have to do with efficiently computing low-rank matrix approximations that are explicitly expressed in terms of a small number of columns and/or rows of the input matrix. We start with the following definition.

DEFINITION 1. $\ldots$ A $\ldots$ $m \times n$ $\ldots$ $\ldots$ C $\ldots$ $m \times c$ $\ldots$ $\ldots$ c $\ldots$ $\ldots$ A $\ldots$ $m \times n$ $\ldots$ $A' = CX$ $\ldots$ column-based matrix approximation $\ldots$ A $\ldots$ CX matrix decomposition $\ldots$ $c \times n$ $\ldots$ X

Several things should be noted about this definition. First, we will be interested in $c \ll n$ in our applications. For example, depending on the application, c could be constant, independent of n, logarithmic in the size of n, or simply a large constant factor less than n. Second, a CX matrix decomposition expresses each of the columns

[1] For example, in their review article "Vector algebra in the analysis of genome-wide expression data" [44], which appeared in *Genome Biology*, Kuruvilla, Park, and Schreiber describe many uses of the vectors provided by the SVD and PCA in DNA microarray analysis. The three biologists then conclude by stating that: "While very efficient basis vectors, the vectors themselves are completely artificial and do not correspond to actual (DNA expression) profiles. ... Thus, it would be interesting to try to find basis vectors for all experiment vectors, using actual experiment vectors and not artificial bases that offer little insight." That is, they explicitly state that they would like decompositions of the form we provide in this paper!

of A in terms of a linear combination of "dictionary elements" or "basis columns," each of which is an actual column of A. Thus, a CX matrix decomposition provides a low-rank approximation to the original matrix, although one with structural properties that are quite different than those provided by the SVD. Third, given a set of columns C, the approximation $A' = P_C A = CC^+A$ (where $P_C A$ is the projection of A onto the subspace spanned by the columns of C and C^+ is the Moore–Penrose generalized inverse of C, as defined in section 2) clearly satisfies the requirements of Definition 1. Indeed, this is the "best" such approximation to A, in the sense that $\|A - C(C^+A)\|_F = \min_{X \in \mathbb{R}^{c \times n}} \|A - CX\|_F$.

Our first main result is the following.

THEOREM 1. [illegible] $A \in \mathbb{R}^{m \times n}$ [illegible] $k \ll \min\{m, n\}$ [illegible] $c = O(k^2 \log(1/\delta)/\epsilon^2)$ [illegible] A [illegible] C [illegible] $c = O(k \log k \log(1/\delta)/\epsilon^2)$ [illegible] C [illegible] $1 - \delta$

$$\min_{X \in \mathbb{R}^{c \times n}} \|A - CX\|_F = \left\|A - CC^+A\right\|_F \leq (1 + \epsilon) \|A - A_k\|_F . \tag{1}$$

[illegible] C [illegible] A CC^+A [illegible] A [illegible] A_k [illegible] k [illegible] A [illegible] $O(SVD(A, k))$ [illegible] k [illegible] A [37]

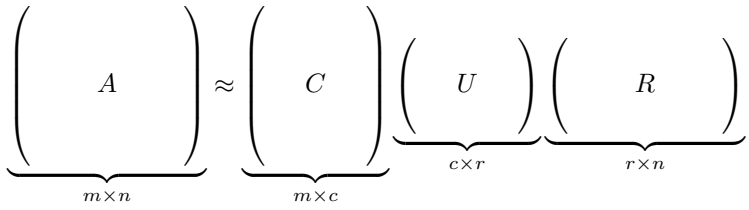
Note that we use $c > k$ and have an ϵ error, which allows us to take advantage of linear algebraic structure in order to obtain an efficient algorithm. In general, this would not be the case if, given an $m \times n$ matrix A, we had specified a parameter k and asked for the "best" subset of k columns, where "best" is measured, e.g., by maximizing the Frobenius norm captured by projecting onto those columns or by maximizing the volume of the parallelepiped defined by those columns. Also, it is not clear a priori that C with properties above even exists; see the discussion in sections 3.2 and 3.3. Finally, our result does not include any reference to regularization or conditioning, as is common in certain application domains; a discussion of similar work on related problems in numerical linear algebra may be found in section 3.1.

Our second main result extends the previous result to [illegible] matrix decompositions.

DEFINITION 2. [illegible] A [illegible] $m \times n$ [illegible] C [illegible] $m \times c$ [illegible] c [illegible] A [illegible] R [illegible] $r \times n$ [illegible] r [illegible] A [illegible] $m \times n$ [illegible] $A' = CUR$ [illegible] column-row-based matrix approximation [illegible] A [illegible] CUR matrix decomposition [illegible] [illegible] $c \times r$ [illegible] U

Several things should be noted about this definition. First, a [illegible] matrix decomposition is a CX matrix decomposition, but one with a very special structure; i.e., every column of A can be expressed in terms of the basis provided by C using only the information contained in a small number of rows of A and a low-dimensional encoding matrix. Second, in terms of its singular value structure, U must clearly contain "inverse-of-A" information. For the [illegible] decomposition described in this paper, U will be a generalized inverse of the intersection between C and R. More precisely, if $C = AS_C D_C$ and $R = D_R S_R^T A$, then $U = (D_R S_R^T A S_C D_C)^+$. (See section 2 for a review of linear algebra and notation, such as that for S_C, D_C, S_R, and D_R.) Third, the combined size of C, U, and R is $O(mc + rn + cr)$, which is an improvement over A's size of $O(mn)$ when $c, r \ll n, m$. Finally, note the structural simplicity of a [illegible]

matrix decomposition:

(2) 

Our main result for matrix decomposition is the following.

THEOREM 2. $A \in \mathbb{R}^{m \times n}$ $k \ll \min\{m, n\}$ $c = O(k^2 \log(1/\delta)/\epsilon^2)$ A C $r = O(c^2 \log(1/\delta)/\epsilon^2)$ A R $c = O(k \log k \log(1/\delta)/\epsilon^2)$ A C $r = O(c \log c \log(1/\delta)/\epsilon^2)$ A R $1 - \delta$

$$\|A - CUR\|_F \leq (1 + \epsilon) \|A - A_k\|_F . \tag{3}$$

U C R A_k k A $O(SVD(A, k))$ k A [37]

1.3. Summary of main technical result. The key technical insight that leads to the relative-error guarantees is that the columns are selected by a novel sampling procedure that we call "subspace sampling." Rather than sample columns from A with a probability distribution that depends on the Euclidean norms of the columns of A (which gives provable additive-error bounds [21, 22, 23]), in "subspace sampling" we randomly sample columns of A with a probability distribution that depends on the Euclidean norms of the rows of the top k right singular vectors of A. This allows us to capture entirely a certain subspace of interest. Let $V_{A,k}$ be the $n \times k$ matrix whose columns consist of the top k right singular vectors of A. The "subspace sampling" probabilities $p_i, i \in [n]$ will satisfy

$$p_i \geq \frac{\beta \left| (V_{A,k})_{(i)} \right|_2^2}{k} \qquad \forall i \in [n], \tag{4}$$

for some $\beta \in (0, 1]$, where $(V_{A,k})_{(i)}$ is the ith of $V_{A,k}$. That is, we will sample based on the norms of the rows (not the columns) of the truncated matrix of singular vectors. Note that $\sum_{j=1}^{n} |(V_{A,k})_{(j)}|_2^2 = k$ and that $\sum_{i \in [n]} p_i = 1$. To construct sampling probabilities satisfying Condition (4), it is sufficient to spend $O(SVD(A, k))$ time to compute (exactly or approximately, in which case $\beta = 1$ or $\beta < 1$, respectively) the top k right singular vectors of A. Sampling probabilities of this form will allow us to deconvolute subspace information and "size-of-A" information in the input matrix A, which in turn will allow us to obtain the relative-error guarantees we desire. Note that we have used this method previously [29], but in that case the sampling probabilities contained other terms that complicated their interpretation.

We will use these "subspace sampling" probabilities in our main technical result, which is a random sampling algorithm for approximating the following generalized

version of the standard ℓ_2 regression problem. Our main column/row-based approximation algorithmic results will follow from this result. Given as input a matrix $A \in \mathbb{R}^{m \times n}$ that has rank no more than k and a matrix of target vectors $B \in \mathbb{R}^{m \times p}$ compute

$$\mathcal{Z} = \min_{X \in \mathbb{R}^{n \times p}} \|B - AX\|_F . \tag{5}$$

That is, fit every column of the matrix B to the basis provided by the columns of the rank-k matrix A. Also of interest is the computation of

$$X_{opt} = A^+ B. \tag{6}$$

The main technical result of this paper is a simple sampling algorithm that represents the matrices A and B by a small number of rows so that this generalized ℓ_2 regression problem can be solved to accuracy $1 \pm \epsilon$ for any $\epsilon > 0$.

More precisely, we present and analyze an algorithm (Algorithm 3 of section 6) that constructs and solves an induced subproblem of the generalized ℓ_2 regression problem of (5) and (6). Let $DS^T A$ be the $r \times n$ matrix consisting of the sampled and appropriately rescaled rows of the original matrix A, and let $DS^T B$ be the $r \times p$ matrix consisting of the sampled and appropriately rescaled rows of B. Then consider the problem

$$\tilde{\mathcal{Z}} = \min_{X \in \mathbb{R}^{n \times p}} \left\|DS^T B - DS^T AX\right\|_F . \tag{7}$$

The "smallest" matrix $\tilde{X}_{opt} \in \mathbb{R}^{n \times p}$ among those that achieve the minimum value $\tilde{\mathcal{Z}}$ in this sampled ℓ_2 regression problem is

$$\tilde{X}_{opt} = \left(DS^T A\right)^+ DS^T B. \tag{8}$$

Since we will sample a number of rows $r \ll m$ of the original problem, we will compute (8), and thus (7), exactly. Our main theorem, Theorem 5, states that under appropriate assumptions on the original problem and on the sampling probabilities, the computed quantities $\tilde{\mathcal{Z}}$ and $\tilde{X}_{opt}$ will provide very accurate relative-error approximations to the exact solution $\mathcal{Z}$ and the optimal vector X_{opt}. Rows will be sampled with one of two random sampling procedures. In one case, exactly $r = O(k^2/\epsilon^2)$ rows are chosen, and in the other case, $r = O(k \log k/\epsilon^2)$ rows in expectation are chosen. In either case, the most expensive part of the computation involves the computation of the Euclidean norms of the rows of the right singular vectors of A which are used in the sampling probabilities.

1.4. Outline of the remainder of the paper. In the next two sections, we provide a review of relevant linear algebra, and we discuss related work. Then, in sections 4 and 5, we present in detail our main algorithmic results. In section 4, we describe our main column-based matrix approximation algorithm, and in section 5, we describe our main column-row-based matrix approximation algorithm. Then, in section 6, we present an approximation algorithm for generalized ℓ_2 regression. This is our main technical result, and from it our two main algorithmic results will follow. Finally, in section 7 we present an empirical evaluation of our algorithms, and in section 8 we present a brief conclusion. We devote Appendix A to two prior algorithms for approximate matrix multiplication. These two algorithms select columns and rows in a complementary manner, and they are essential in the proof of our main results.

2. Review of linear algebra. In this section, we provide a review of linear algebra that will be useful throughout the paper; for more details, see [50, 43, 60, 37, 9, 6]. We also review a sampling matrix formalism that will be convenient in our discussion [21].

Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For any matrix $A \in \mathbb{R}^{m \times n}$, let $A_{(i)}, i \in [m]$ denote the ith row of A as a row vector, and let $A^{(j)}, j \in [n]$ denote the jth column of A as a column vector. In addition, let $\|A\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2$ denote the square of its Frobenius norm, and let $\|A\|_2 = \sup_{x \in \mathbb{R}^n,\ x \neq 0} |Ax|_2 / |x|_2$ denote its spectral norm. These norms satisfy $\|A\|_2 \leq \|A\|_F \leq \sqrt{\min\{m,n\}}\, \|A\|_2$ for any matrix A, and also $\|AB\|_F \leq \|A\|_F \, \|B\|_2$ for any matrices A and B.

If $A \in \mathbb{R}^{m \times n}$, then there exist orthogonal matrices $U = [u^1 u^2 \ldots u^m] \in \mathbb{R}^{m \times m}$ and $V = [v^1 v^2 \ldots v^n] \in \mathbb{R}^{n \times n}$ such that $U^T A V = \Sigma = \mathbf{diag}(\sigma_1, \ldots, \sigma_\xi)$, where $\Sigma \in \mathbb{R}^{m \times n}$, $\xi = \min\{m, n\}$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_\xi \geq 0$. Equivalently, $A = U \Sigma V^T$. The three matrices U, V, and Σ constitute the SVD of A. The σ_i are the singular values of A, the vectors u^i and v^i are the ith left and the ith right singular vectors of A, respectively, and the condition number of A is $\kappa(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$. If $k \leq r = \mathrm{rank}(A)$, then the SVD of A may be written as

(9)
$$A = U_A \Sigma_A V_A^T = \left[\begin{array}{cc} U_k & U_k^\perp \end{array} \right] \left[\begin{array}{cc} \Sigma_k & \mathbf{0} \\ \mathbf{0} & \Sigma_{k,\perp} \end{array} \right] \left[\begin{array}{c} V_k^T \\ V_k^{\perp T} \end{array} \right] = U_k \Sigma_k V_k^T + U_k^\perp \Sigma_{k,\perp} V_k^{\perp T}.$$

Here, Σ_k is the $k \times k$ diagonal matrix containing the top k singular values of A, and $\Sigma_{k,\perp}$ is the $(r-k) \times (r-k)$ diagonal matrix containing the bottom $r-k$ nonzero singular values of A. Also, V_k^T is the $k \times n$ matrix whose rows are the top k right singular vectors of A, $V_k^{\perp T}$ is the $(r-k) \times n$ matrix whose rows are the bottom $r-k$ right singular vectors of A, and U_k and $U_k^\perp$ are defined similarly. If we define $A_k = U_k \Sigma_k V_k^T$, then the distance (as measured by both $\|\cdot\|_2$ and $\|\cdot\|_F$) between A and any rank k approximation to A is minimized by A_k. We will denote by $O(SVD(A,k))$ the time required to compute the best rank-k approximation to the matrix A [37]. Finally, for any orthogonal matrix $U \in \mathbb{R}^{m \times c}$, let $U^\perp \in \mathbb{R}^{m \times (m-c)}$ denote an orthogonal matrix whose columns are an orthonormal basis spanning the subspace of $\mathbb{R}^m$ that is orthogonal to the column space of U.

Given a matrix $A \in \mathbb{R}^{m \times n}$, the unweighted Moore–Penrose generalized inverse of A, denoted by A^+, is the unique $n \times m$ matrix that satisfies the four Moore–Penrose conditions [50, 6]. In terms of the SVD this generalized inverse may be written as $A^+ = V_A \Sigma_A^{-1} U_A^T$ (where the square diagonal $\mathrm{rank}(A) \times \mathrm{rank}(A)$ matrix Σ_A, as in (9), is invertible by construction). If, in addition, $D_1 \in \mathbb{R}^{m \times m}$ and $D_2 \in \mathbb{R}^{n \times n}$ are diagonal matrices with positive entries along the diagonal, then the $\{D_1, D_2\}$-Moore–Penrose generalized inverse of A, denoted by $A^+_{(D_1,D_2)}$, is a generalization of the Moore–Penrose inverse that can be expressed in terms of the unweighted generalized inverse of A as $A^+_{(D_1,D_2)} = D_2^{-1/2} (D_1^{1/2} A D_2^{-1/2})^+ D_1^{1/2}$. Also, in terms of the generalized inverse, the projection onto the column space of any matrix A may be written as $P_A = AA^+$.

Since our main algorithms will involve sampling columns and/or rows from input matrices (using one of two related random sampling procedures described in Appendix A), we conclude this subsection with a brief review of a sampling matrix formalism that was introduced in [21] and with respect to which our sampling matrix operations may be conveniently expressed. First, assume that c' (= c , , ,) columns of A are chosen in c i.i.d. trials by randomly sampling according to

a probability distribution $\{p_i\}_{i=1}^n$ with the EXACTLY(c) algorithm (described in detail in Appendix A), and assume that the i_tth column of A is chosen in the tth (for $t = 1, \ldots, c$) independent random trial. Then, define the sampling matrix $S \in \mathbb{R}^{n \times c}$ to be the zero-one matrix where $S_{i_t t} = 1$ and $S_{ij} = 0$ otherwise, and define the diagonal rescaling matrix $D \in \mathbb{R}^{c \times c}$ to be the diagonal matrix with $D_{tt} = 1/\sqrt{cp_{i_t}}$, where p_{i_t} is the probability of choosing the i_tth column. Alternatively, assume that c' ($\leq c$, with high probability) columns of A are chosen with the EXPECTED(c) algorithm (also described in detail in Appendix A) by including the ith column of A in C with probability $\tilde{p}_i = \min\{1, cp_i\}$. Then, define the sampling matrix $S \in \mathbb{R}^{n \times n}$ to be the zero-one matrix where $S_{ii} = 1$ if the ith column is chosen and $S_{ij} = 0$ otherwise, and define the rescaling matrix $D \in \mathbb{R}^{n \times c'}$ to be the matrix with $D_{ij} = 1/\sqrt{c\tilde{p}_j}$ if $i - 1$ of the previous columns have been chosen and $D_{ij} = 0$ otherwise. Clearly, in both of these cases, $C = ASD$ is an $m \times c'$ matrix consisting of sampled and rescaled copies of the columns of A, and $R = (SD)^T A = DS^T A$ is a $c' \times n$ matrix consisting of sampled and rescaled copies of the rows of A. In certain cases, we will subscript S and D with C or R (e.g., $C = AS_C D_C$ and $R = D_R S_R^T A$) to make explicit that the corresponding sampling and rescaling matrices are operating on the columns or rows, respectively, of A.

3. Relationship with previous related work. In this section, we discuss the relationship between our results and related work in numerical linear algebra and theoretical computer science.

3.1. Related work in numerical linear algebra. Within the numerical linear algebra community, several groups have studied matrix decompositions with similar structural, if not algorithmic, properties to the CX and , matrix decompositions we have defined. Much of this work is related to the QR decomposition, originally used extensively in pivoted form by Golub [36, 11].

Stewart and collaborators were interested in computing sparse low-rank approximations to large sparse term-document matrices [58, 59, 8]. He developed the quasi-Gram–Schmidt method. This method is a variant of the QR decomposition which, when given as input an $m \times n$ matrix A and a rank parameter k, returns an $m \times k$ matrix C consisting of k columns of A whose span approximates the column space of A and also a nonsingular upper-triangular $k \times k$ matrix T_C that orthogonalizes these columns (but it does not explicitly compute the nonsparse orthogonal matrix $Q_C = CT_C^{-1}$). This provides a matrix decomposition of the form $A \approx CX$. By applying this method to A to obtain C and to A^T to obtain an $k \times n$ matrix R consisting of k rows of A, one can show that $A \approx CUR$, where the matrix U is computed to minimize $\|A - CUR\|_F^2$. Although provable approximation guarantees of the form we present were not provided, backward error analysis was performed, and the method was shown to perform well empirically [58, 59, 8].

Goreinov, Tyrtyshnikov, and Zamarashkin [39, 38, 61] were interested in applications such as scattering, in which large coefficient matrices have blocks that can be easily approximated by low-rank matrices. They show that if the matrix A is approximated by a rank-k matrix to within an accuracy ϵ, then , a choice of k columns and k rows, i.e., C and R, and a low-dimensional $k \times k$ matrix U constructed from the elements of C and R, such that $A \approx CUR$ in the sense that $\|A - CUR\|_2 \leq \epsilon f(m, n, k)$, where $f(m, n, k) = 1 + 2\sqrt{km} + 2\sqrt{kn}$. In [39], the choice for these matrices is related to the problem of determining the minimum singular value σ_k of $k \times k$ submatrices of $n \times k$ orthogonal matrices. In addition, in [38]

the choice for C and R is interpreted in terms of the maximum volume concept from interpolation theory, in the sense that columns and rows should be chosen such that their intersection W defines a parallelepiped of maximum volume among all $k \times k$ submatrices of A; in [61] an empirically effective deterministic algorithm is presented which ensures that U is well-conditioned.

Gu and Eisenstat, in their seminal paper [40], describe a strong rank-revealing QR factorization that deterministically selects exactly k columns from an $m \times n$ matrix A. The algorithms of [40] are efficient, in that their running time is $O(mn^2)$ (assuming that $m \geq n$), which is essentially the time required to compute the SVD of A. In addition, Gu and Eisenstat prove that if the $m \times k$ matrix C contains the k selected columns (without any rescaling), then $\sigma_{\min}(C) \geq \sigma_k(A)/f(k,n)$, where $f(k,n) = O(\sqrt{k(n-k)})$. Thus, the columns of C span a parallelepiped whose volume (equivalently, the product of the singular values of C) is "large." Currently, we do not know how to convert this property into a statement similar to that of Theorem 1, although perhaps this can be accomplished by relaxing the number of columns selected by the algorithms of [40] to $O(poly(k, 1/\epsilon))$. For related work prior to Gu and Eisenstat, see Chan and Hansen [12, 13].

Finally, very recently, Martinsson, Rokhlin, and Tygert [49] proposed another related method to efficiently compute an approximation to the best rank-k approximation of an $m \times n$ matrix A. The heart of their algorithm is a random projection method, which projects A to a small number, say ℓ, of random vectors; the entries of these random vectors are i.i.d. Gaussians of zero mean and unit variance. The general form of their bounds is quite complicated, but by setting, e.g., $\ell = k + 20$, they construct a rank-k approximation A' to A such that

$$\|A - A'\|_2 \leq 10\sqrt{(k+20)\,m}\,\|A - A_k\|_2 \tag{10}$$

holds with probability at least $1 - 10^{-17}$. In addition, the authors extend their algorithm to compute the so-called interpolative decomposition of a matrix A. This decomposition is explicitly expressed in terms of a small number of columns of A, and is a more restrictive version of our CX matrix decomposition. More specifically, it additionally requires that every entry of X is bounded in absolute value by a small constant (e.g., two). Thus, their algorithm computes an interpolative approximation $A' = CX$ to A, where C has only $\ell = k + 20$ columns—as opposed to the $O(k \log k)$ columns that are necessary in our work—and satisfies the bound of (10). Notice that their work provides bounds for the spectral norm, whereas our work focuses only on the Frobenius norm. However, their bounds are much weaker than our relative error bounds, since $\sqrt{m(k+20)}\,\|A - A_k\|_2$ might in general be larger even than $\|A\|_F$.

3.2. Related work in theoretical computer science. Within the theory of algorithms community, much research has followed the seminal work of Frieze, Kannan, and Vempala [33, 34]. Their work may be viewed, in our parlance, as sampling columns from a matrix A to form a matrix C such that $\|A - CX\|_F \leq \|A - A_k\|_F + \epsilon\,\|A\|_F$. The matrix C has $poly(k, 1/\epsilon, 1/\delta)$ columns and is constructed after making only two passes over A using $O(m+n)$ work space. Under similar resource constraints, a series of papers have followed [33, 34] in the past seven years [19, 22, 55], improving the dependency of c on $k, 1/\epsilon$, and $1/\delta$, and analyzing the spectral as well as the Frobenius norm, yielding bounds of the form

$$\|A - CX\|_\xi \leq \|A - A_k\|_\xi + \epsilon\,\|A\|_F \tag{11}$$

for $\xi = 2, F$, and thus providing additive-error guarantees for column-based low-rank matrix approximations.

Additive-error approximation algorithms for , matrix decompositions have also been analyzed by Drineas, Kannan, and Mahoney [20, 21, 22, 23, 24, 25]. In particular, in [23], they compute an approximation to an $m \times n$ matrix A by sampling c columns and r rows from A to form $m \times c$ and $r \times n$ matrices C and R, respectively. From C and R, a $c \times r$ matrix U is constructed such that under appropriate assumptions

$$\|A - CUR\|_\xi \leq \|A - A_k\|_\xi + \epsilon \|A\|_F , \tag{12}$$

with high probability, for both the spectral and Frobenius norms, $\xi = 2, F$. In [24, 25], it is further shown that if A is a symmetric positive semidefinite (SPSD) matrix, then one can choose $R = C^T$ and $U = W^+$, where W is the $c \times c$ intersection between C and $R = C^T$, thus obtaining an approximation $A \approx A' = CW^+C^T$. This approximation is SPSD and has provable bounds of the form (12), except that the scale of the additional additive error is somewhat larger [24, 25].

Most relevant for our relative-error CX and , matrix decomposition algorithms is the recent work of Rademacher, Vempala, and Wang [53] and Deshpande, Rademacher, Vempala, and Wang [17]. Using two different methods (in one case iterative sampling in a backwards manner and an induction on k argument [53], and in the other case an argument that relies on estimating the volume of the simplex formed by each of the k-sized subsets of the columns [17]), they reported the of a set of $O(k^2/\epsilon^2)$ columns that provide relative-error CX matrix decomposition. No algorithmic result was presented, except for an exhaustive algorithm that ran in $\Omega(n^k)$ time. Note that their results did not apply to columns and rows . Thus, ours is the first , matrix decomposition algorithm with relative error, and it was previously not even known whether such a relative-error CUR representation existed; i.e., it was not previously known whether columns and rows satisfying the conditions of Theorem 2 existed.

Other related work includes that of Rudelson and Vershynin [54, 62, 56], who provide an algorithm for CX matrix decomposition which has an improved additive error spectral norm bound of the form

$$\|A - CX\|_2 \leq \|A - A_k\|_2 + \epsilon \sqrt{\|A\|_2 \, \|A\|_F} .$$

Their proof uses an elegant result on random vectors in the isotropic position [54], and since we use a variant of their result, it is described in more detail in Appendix A. Achlioptas and McSherry have computed low-rank matrix approximations using sampling techniques that involve zeroing-out and/or quantizing individual elements [2, 1]. The primary focus of their work was in introducing methods to accelerate orthogonal iteration and Lanczos iteration methods, and their analysis relied heavily on ideas from random matrix theory [2, 1]. Agarwal, Har-Peled, and Varadarajan have analyzed so-called "core sets" as a tool for efficiently approximating various extent measures of a point set [3, 4]. The choice of columns and/or rows we present are a "core set" for approximate matrix computations; in fact, our algorithmic solution to Theorem 1 solves an open question in their survey [4]. The choice of columns and rows we present may also be viewed as a set of variables and features chosen from a data matrix [10, 14, 41]. "Feature selection" is a broad area that addresses the choice of columns explicitly for dimension reduction, but the metrics there are typically optimization based [14] or machine-learning based [10]. These formulations tend to have set cover-like solutions

and are incomparable with the linear-algebraic structure such as the low-rank criteria we consider here that is common among data analysts.

3.3. Very recent work on relative-error approximation algorithms. To the best of our knowledge, the first nontrivial [illegible] for relative-error low-rank matrix approximation was provided by a preliminary version of this paper [27, 28]. In particular, an earlier version of Theorem 1 provided the first known relative-error, column-based, low-rank approximation in polynomial time [27, 28]. The major difference between our Theorem 1 and our result in [27, 28] is that the sampling probabilities in [27, 28] are more complicated. (See section 6.2 for details on this.) The algorithm from [27, 28] runs in $O(SVD(A,k))$ time (although it was originally reported to run in only $O(SVD(A))$ time), and it has a sampling complexity of $O(k^2 \log(1/\delta)/\epsilon^2)$ columns.

Subsequent to the completion of the preliminary version of this paper [27, 28], several developments have been made on relative-error low-rank matrix approximation algorithms. First, Har-Peled reported an algorithm that takes as input an $m \times n$ matrix A, and in roughly $O(mnk^2 \log k)$ time returns as output a rank-k matrix A' with a relative-error approximation guarantee [42]. His algorithm uses geometric ideas and involves sampling and merging approximately optimal k-flats; it is not clear if this approximation can be expressed in terms of a small number of columns of A. Then, Deshpande and Vempala [18] reported an algorithm that takes as input an $m \times n$ matrix A that also returns a relative-error approximation guarantee. Their algorithm extends ideas from [53, 17], and it leads to a CX matrix decomposition consisting of $O(k \log k)$ columns of A. The complexity of their algorithm is $O(Mk^2 \log k)$, where M is the number of nonzero elements of A, and their algorithm can be implemented in a data-streaming framework with $O(k \log k)$ passes over the data. In light of these developments, we simplified and generalized our preliminary results [27, 28], and we performed a more refined analysis to improve our sampling complexity to $O(k \log k)$. Most recently, we learned of work by Sarlos [57], who used ideas from the recently developed fast Johnson–Lindenstrauss transform of Ailon and Chazelle [5] to yield further improvements to a CX matrix decomposition.

4. Our main column-based matrix approximation algorithm. In this section, we describe an algorithm and a theorem, from which our first main result, Theorem 1, will follow.

4.1. Description of the algorithm. Algorithm 1 takes as input an $m \times n$ matrix A, a rank parameter k, and an error parameter ϵ. It returns as output an $m \times c$ matrix C consisting of a small number of columns of A. The algorithm is very simple: sample a small number of columns according to a carefully constructed nonuniform probability distribution. Algorithm 1 uses the sampling probabilities

$$p_i = \frac{1}{k} \left| \left(V_{A,k}^T\right)^{(i)} \right|_2^2, \qquad \forall i \in [n], \tag{13}$$

but it will be clear from the analysis of section 6 that any sampling probabilities such that $p_i \geq \beta |(V_{A,k}^T)^{(i)}|_2^2/k$, for some $\beta \in (0,1]$, will also work with a small β-dependent loss in accuracy. Note that Algorithm 1 actually consists of two related algorithms, depending on how exactly the columns are chosen. The EXACTLY(c) algorithm picks exactly c columns of A to be included in C in c i.i.d. trials, where in each trial the ith column of A is picked with probability p_i. The EXPECTED(c) algorithm picks in expectation at most c columns of A to create C, by including the ith column of A

in C with probability $\min\{1, cp_i\}$. See Algorithms 4 and 5 in Appendix A for more details about these two column-sampling procedures.

Data : $A \in \mathbb{R}^{m\times n}$, a rank parameter k, and an error parameter ϵ.
Result : $C \in \mathbb{R}^{m\times c}$

- Compute sampling probabilities p_i for all $i \in [n]$ given by (13);
- (Implicitly) construct a sampling matrix S_C and a diagonal rescaling matrix D_C with the EXACTLY(c) algorithm or with the EXPECTED(c) algorithm;
- Construct and return the matrix $C = AS_CD_C$ consisting of a small number of rescaled columns of A.

Algorithm 1. A randomized algorithm for CX matrix decomposition.

The running time of Algorithm 1 is dominated by the computation of the sampling probabilities (13), for which $O(SVD(A,k))$ time suffices. The top k right singular vectors of A can be efficiently (approximately) computed using standard algorithms [37, 51]. The building block of these algorithms is a series of matrix-vector multiplications, where the input matrix A is iteratively multiplied with a changing set of k orthogonal vectors. In each iteration (which can be implemented by making passes over the input matrix A), the accuracy of the approximation improves. Even though the number of iterations required to bound the error depends on quantities such as the gap between the singular values of A, these algorithms work extremely well in practice. As such, they are often treated as "black boxes" for SVD computation in the TCS literature; see, e.g., [2, 1].

4.2. Statement of the theorem. Theorem 3 is our main quality-of-approximation result for Algorithm 1.

THEOREM 3. Let $A \in \mathbb{R}^{m\times n}$, let k [illegible] $\epsilon \in (0,1]$ [illegible] $c = 3200k^2/\epsilon^2$ [illegible] 1 [illegible] c [illegible] A [illegible] EXACTLY(c) [illegible] 0.7

$$\left\|A - CC^+A\right\|_F \le (1+\epsilon)\left\|A - A_k\right\|_F. \tag{14}$$

[illegible] $c = O(k\log k/\epsilon^2)$ [illegible] 1 [illegible] c [illegible] A [illegible] EXPECTED(c) [illegible] (14) [illegible] 0.7

[illegible]. Since for every set of columns $C = AS_CD_C$, $X_{opt} = C^+A$ is the matrix that minimizes $\|A - CX\|_F$, it follows that

$$\begin{aligned}\left\|A - CC^+A\right\|_F &= \left\|A - (AS_CD_C)(AS_CD_C)^+A\right\|_F \\ &\le \left\|A - (AS_CD_C)(P_{A,k}AS_CD_C)^+P_{A,k}A\right\|_F,\end{aligned} \tag{15}$$

where $P_{A,k} = U_{A,k}U_{A,k}^T$ is a projection onto the top k left singular vectors of A. To bound (15), consider the problem of approximating the solution to $\min_{X\in\mathbb{R}^{m\times m}} \|XA_k - A\|_F$ by randomly sampling columns of A_k and of A. It follows as a corollary of (21) of Theorem 5 of section 6 that

$$\left\|A - (AS_CD_C)(A_kS_CD_C)^+A_k\right\|_F \le (1+\epsilon)\left\|A - AA_k^+A_k\right\|_F = (1+\epsilon)\left\|A - A_k\right\|_F, \tag{16}$$

which, when combined with (15), establishes the theorem. □

For simplicity of presentation, we have presented Algorithm 1 and Theorem 3 such that (14) holds with only constant probability, but this can be boosted to hold with probability at least $1-\delta$ using standard methods. In particular, consider the following: run Algorithm 1 (using either the EXACTLY(c) algorithm or the EXPECTED(c) algorithm, but with the appropriate value of c) independently $\ln(1/\delta)$ times, and return the C such that $\|A - CC^+A\|_F$ is smallest. Then, since in each trial the claim of Theorem 3 fails with probability less than $0.3 < 1/e$, the claim of Theorem 3 will fail for every trial with probability less than $(1/e)^{\ln(1/\delta)} = \delta$. This establishes Theorem 1.

For simplicity of presentation, we have also stated Theorem 3 in such a way that the rank of the approximating matrix $A' = CC^+A$ may be greater than k. This possibility may be undesirable in certain applications, and it can be easily removed. Let $A'' = C(P_{A,k}C)^+P_{A,k}A$. Then, it follows from (16) that A'' is a CX matrix approximation that is within relative error ϵ of the best rank-k approximation to A and that has rank no more than k.

4.3. Discussion of the analysis. Given a matrix A, Theorem 1 asks us to find a set of columns $C = AS_CD_C$ such that CC^+A "captures" almost as much of A as does $A_k = U_{A,k}U_{A,k}^TA$. Given that set (or any other set) of columns C, it is well known that the matrix $X_{opt} = C^+A$ is the "smallest" matrix among those that solve the optimization problem (19). For a given A and C, let us approximate X_{opt} as

$$X_{opt} = C^+A \approx (P_{A,k}C)^+ P_{A,k}A.$$

This approximation is suboptimal with respect to solving the optimization problem (19), i.e.,

$$\left\|A - CC^+A\right\|_F \leq \left\|A - C\left(P_{A,k}C\right)^+ P_{A,k}A\right\|_F,$$

but it can be shown that by choosing C properly, i.e., by choosing S_C and D_C (the column sampling and rescaling matrices) properly, we have that

$$\left\|A - C\left(P_{A,k}C\right)^+ P_{A,k}A\right\|_F \leq (1+\epsilon)\left\|A - A_k\right\|_F.$$

The main technical challenge is to sample in a manner such that the column-sampled version of the matrix consisting of the top k right singular vectors of A is full rank; i.e., $\mathrm{rank}(V_{A,k}^TS_CD_C) = \mathrm{rank}(V_{A,k}^T) = k$. To accomplish this, we sample with respect to probabilities of the form (13). To understand these sampling probabilities, recall that we seek to pick columns that span almost the same subspace as the top k left singular vectors of A (i.e., U_k), and recall that the ith column of A is equal to

$$A^{(i)} = U_k\Sigma_k\left(V_k^T\right)^{(i)} + U_{\rho-k}\Sigma_{\rho-k}\left(V_{\rho-k}^T\right)^{(i)}.$$

Since postmultiplying U_k by Σ_k does not change the span of the columns of U_k, $|(V_k^T)^{(i)}|_2^2$ measures "how much" of the ith column of A lies in the span of $U_{A,k}$, independent of the magnitude of the singular values associated with those directions.

5. Our main column-row-based matrix approximation algorithm. In this section, we describe an algorithm and a theorem that, when combined with the results of section 4, will establish our second main result, Theorem 2.

5.1. Description of the algorithm. Algorithm 2 takes as input an $m \times n$ matrix A, an $m \times c$ matrix C consisting of a small number of columns of A, and an error parameter ϵ. It returns as output an $r \times n$ matrix R consisting of a small number of rows of A and an $r \times c$ matrix W consisting of the corresponding rows of C. The algorithm is very simple: sample a small number of rows according to a carefully constructed nonuniform probability distribution. Algorithm 2 uses the sampling probabilities

$$p_i = \frac{1}{c}\left|\left(U_C^T\right)^{(i)}\right|_2^2, \qquad \forall i \in [m], \tag{17}$$

but it will be clear from the analysis of section 6 that any sampling probabilities $p_i, i \in [m]$, such that $p_i \geq \beta|(U_C^T)^{(i)}|_2^2/c$, for some $\beta \in (0,1]$, will also work with a small β-dependent loss in accuracy. Note that Algorithm 2 actually consists of two related algorithms, depending on how exactly the rows are chosen. The EXACTLY(c) algorithm picks exactly r rows of A to be included in R in r i.i.d. trials, where in each trial the ith row of A is picked with probability p_i. The EXPECTED(c) algorithm picks in expectation at most r rows of A to create R, by including the ith column of A in C with probability $\min\{1, rp_i\}$. See Algorithms 4 and 5 in Appendix A for more details about these two row-sampling procedures.

Data : $A \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{m\times c}$ consisting of c columns of A, a positive integer r, and an error parameter ϵ.

Result : $R \in \mathbb{R}^{r\times n}$ consisting of r rows of A and $W \in \mathbb{R}^{c\times r}$ consisting of the corresponding r rows of C, and $U \in \mathbb{R}^{r\times c}$.

- Compute sampling probabilities p_i for all $i \in [m]$ given by (17);
- (Implicitly) construct a sampling matrix S_R and a diagonal rescaling matrix D_R with the EXACTLY(c) algorithm or with the EXPECTED(c) algorithm;
- Construct and return the matrix $R = D_R S_R^T A$ consisting of a small number of rescaled rows of A;
- Construct and return the matrix $W = D_R S_R^T C$ consisting of the corresponding rescaled rows of C;
- Let $U = W^+$.

Algorithm 2. A randomized algorithm for *CUR* matrix decomposition.

Reading the input matrices to Algorithm 2 takes $O(mn)$ time; computing the full SVD of C requires $O(c^2m)$ time; constructing the matrix R requires $O(rn)$ time; constructing the matrix W requires $O(rc)$ time; and computing U requires $O(c^2r)$ time. Overall, the running time of the algorithm is $O(mn)$ since c, r are constants independent of m, n. This can be improved if the input matrices are sparse, but for simplicity we omit this discussion.

5.2. Statement of the theorem. Theorem 4 is our main quality-of-approximation result for Algorithm 2.

THEOREM 4. Let $A \in \mathbb{R}^{m\times n}$ and $C \in \mathbb{R}^{m\times c}$ [illegible] c [illegible] A [illegible] $\epsilon \in (0,1]$ [illegible] $r = 3200c^2/\epsilon^2$ [illegible] 2 [illegible] r [illegible] A [illegible] C [illegible] EXACTLY(c) [illegible]

0.7

$$\|A - CUR\|_F \leq (1+\epsilon)\left\|A - CC^+A\right\|_F. \tag{18}$$

$r = O(c \log c/\epsilon^2)$ 2 r A C Expected(c) (18)

0.7

Consider the problem of approximating the solution to $\min_{X \in \mathbb{R}^{c \times n}} \|CX - A\|_F$ by randomly sampling rows from C and A. It follows as a corollary of (21) of Theorem 5 of section 6 that

$$\left\|A - C(D_R S_R^T C)^+ D_R S_R^T A\right\|_F \leq (1+\epsilon)\left\|A - CC^+A\right\|_F,$$

where $R = D_R S_R^T A$ and $U = (D_R S_R^T C)^+$, which establishes the theorem. □

For simplicity of presentation, we have presented Algorithm 2 and Theorem 4 such that (18) holds with only constant probability, but this can be boosted to hold with probability at least $1 - \delta$ using standard methods. In addition, this can be combined with Algorithm 1 and Theorem 3 by doing the following: run Algorithm 1 $\ln(2/\delta)$ times, and return the best C; then, with that C run Algorithm 2 $\ln(2/\delta)$ times, and return the best U, R pair. Then

$$\|A - CUR\|_F \leq (1+\epsilon)\left\|A - CC^+A\right\|_F \leq (1+\epsilon)^2 \|A - A_k\|_F \leq (1+\epsilon')\|A - A_k\|_F,$$

where $\epsilon' = 3\epsilon$, and the combined failure probability is no more than $\delta/2 + \delta/2 = \delta$. This establishes Theorem 2.

5.3. Discussion of the analysis. Assume that we are given an $m \times c$ matrix C, consisting of any set of c columns of an $m \times n$ matrix A, and consider the following idea for approximating the matrix A. The columns of C are a set of "basis vectors" that are, in general, neither orthogonal nor normal. To express all the columns of A as linear combinations of the columns of C, we can solve

$$\min_{x_j \in \mathbb{R}^c} \left|A^{(j)} - Cx_j\right|_2,$$

for each column $A^{(j)}, j \in [n]$, in order to find a c-vector of coefficients x_j and get the optimal least-squares fit for $A^{(j)}$. Equivalently, we can solve an optimization problem of the form (19). Note that if m and n are large and $c = O(1)$, then this is an overconstrained least-squares fit problem. It is well known that $X_{opt} = C^+A$ is the "smallest" matrix solving this optimization problem, in which case we are using information from every row of A to compute the optimal coefficient matrix. Let us approximate X_{opt} as

$$X_{opt} = C^+A \approx \left(D_R S_R^T C\right)^+ D_R S_R^T A = \tilde{X}_{opt},$$

and note that $\tilde{X}_{opt} = W^+R$. This matrix $\tilde{X}_{opt}$ is clearly suboptimal with respect to solving the optimization problem (19), i.e.,

$$\left\|A - CC^+A\right\|_F \leq \left\|A - CW^+R\right\|_F,$$

but it can be shown that by choosing S_R and D_R (the row sampling and rescaling matrices) properly we have that

$$\left\|A - CW^+R\right\|_F \leq (1+\epsilon)\left\|A - CC^+A\right\|_F.$$

As in section 4.3, the main technical challenge is to sample in a manner such that the row-sampled version of the matrix consisting of the top c left singular vectors of C is full rank, i.e., $\text{rank}(D_R S_R^T U_{C,c}) = \text{rank}(U_{C,c}) = c$.

6. An approximation algorithm for generalized ℓ_2 regression. The basic linear-algebraic problem of ℓ_2 regression is one of the most fundamental regression problems, and it has found many applications in mathematics and statistical data analysis. Recall the standard ℓ_2 regression (or least-squares fit) problem: given as input a matrix $A \in \mathbb{R}^{m\times n}$ and a target vector $b \in \mathbb{R}^m$, compute $\mathcal{Z} = \min_{x\in\mathbb{R}^n} |b - Ax|_2$. Also of interest is the computation of vectors that achieve the minimum $\mathcal{Z}$. If $m > n$ there are more constraints than variables and the problem is an overconstrained least-squares fit problem; in this case, there does not in general exist a vector x such that $Ax = b$. It is well known that the minimum-length vector among those minimizing $|b - Ax|_2$ is $x_{opt} = A^+b$. We previously presented an elaborate sampling algorithm that represents the matrix A by a matrix by a small number of rows so that this ℓ_2 regression problem can be solved to accuracy $1 \pm \epsilon$ for any $\epsilon > 0$ [29].

This problem is of interest for CX and , matrix decomposition for the following reason. Given a matrix A and a set of its columns C, if we want to get the best fit for every column of A in terms of that basis, we want to solve $CX \approx A$ for the matrix X. More precisely, we would like to solve the optimization problem such as

$$\mathcal{Z} = \min_{X\in\mathbb{R}^{c\times n}} \|A - CX\|_F . \tag{19}$$

It is well known that the matrix $X = C^+A$ is the "smallest" matrix among those that solve this problem. In this case, we are approximating the matrix A as $A' = CC^+A = P_CA$, and by keeping only the columns C we are incurring an error of $\|A - CC^+A\|_F$. Two questions arise:

- First, how do we choose the columns C such that $\|A - CC^+A\|_F$ is within relative error ϵ of $\|A - A_k\|_F$?
- Second, how do we choose the rows R and a matrix U such that $\|A - CUR\|_F$ is within relative error ϵ of $\|A - CC^+A\|_F$?

Motivated by these observations, we will consider the generalized version of the standard ℓ_2 regression problem, as defined in (5) and (6).

In this section, we first present Algorithm 3, which is our main random sampling algorithm for approximating the solution to the generalized ℓ_2 regression problem, and Theorem 5, which provides our main quality-of-approximation bound for Algorithm 3. Then, we discuss the novel nonuniform "subspace sampling" probabilities used by the algorithm. Finally, we present the proof of Theorem 5.

6.1. Description of the algorithm and theorem. Algorithm 3 takes as input an $m \times n$ matrix A with rank no greater than k, an $m \times p$ matrix B, a set of sampling probabilities $\{p_i\}_{i=1}^m$, and a positive integer $r \le m$. It returns as output a number $\tilde{\mathcal{Z}}$ and a $n \times p$ matrix $\tilde{X}_{opt}$. Using the sampling matrix formalism described in section 2, the algorithm (implicitly) forms a sampling matrix S, the transpose of which samples a few rows of A and the corresponding rows of B, and a rescaling matrix D, which is a matrix scaling the sampled rows of A and B. Since r rows of A and the corresponding r rows of B are sampled, the algorithm randomly samples r of the m constraints in the original ℓ_2 regression problem. Thus, the algorithm approximates the solution of the regression problem $AX \approx B$, as formalized in (19) and (5), with the exact solution of the downsampled regression problem $DS^TAX \approx DS^TB$. Note that it is the space of constraints that is sampled and that the dimensions of the unknown matrix X are the same in both problems. Note also that although both m and n are permitted to be large, the problem is effectively overconstrained since $\mathrm{rank}(A) \le k$. As we will see below, $r = O(k \log k)$ or $r = O(k^2)$, depending on exactly how the random sample is constructed. Thus, we will compute the solution to the sampled problem exactly.

Data : $A \in \mathbb{R}^{m \times n}$ that has rank no greater than k, $B \in \mathbb{R}^{m \times p}$, sampling probabilities $\{p_i\}_{i=1}^m$, and $r \leq m$.

Result : $\tilde{X}_{opt} \in \mathbb{R}^{n \times p}$, $\tilde{\mathcal{Z}} \in \mathbb{R}$.

- (Implicitly) construct a sampling matrix S and a diagonal rescaling matrix D with the EXACTLY(c) algorithm or with the EXPECTED(c) algorithm;
- Construct the matrix DS^TA consisting of a small number of rescaled rows of A;
- Construct the matrix DS^TB consisting of a small number of rescaled rows of B;
- $\tilde{X}_{opt} = \left(DS^TA\right)^+ DS^TB$;
- $\tilde{\mathcal{Z}} = \min_{X \in \mathbb{R}^{n \times p}} \left\| DS^TB - DS^TA\tilde{X}_{opt} \right\|_F$.

Algorithm 3. A Monte-Carlo algorithm for approximating ℓ_2 regression.

Theorem 5 is our main quality-of-approximation result for Algorithm 3. Its proof may be found in section 6.3. Recall that for our generalized ℓ_2 regression problem, the matrix A has rank no greater than k.

THEOREM 5. [illegible] $A \in \mathbb{R}^{m \times n}$ [illegible] k, $B \in \mathbb{R}^{m \times p}$, $\epsilon \in (0,1]$, [illegible] $\mathcal{Z} = \min_{X \in \mathbb{R}^{n \times p}} \|B - AX\|_F = \|B - AX_{opt}\|_F$ [illegible] $X_{opt} = A^+B = A_k^+B$ [illegible] 3 [illegible]

$$p_i \geq \beta \frac{\left|(U_{A,k})_{(i)}\right|_2^2}{\sum_{j=1}^n \left|(U_{A,k})_{(j)}\right|_2^2} = \frac{\beta}{k}\left|(U_{A,k})_{(i)}\right|_2^2, \qquad \forall i \in [n], \tag{20}$$

[illegible] $\beta \in (0,1]$ [illegible] $\tilde{\mathcal{Z}}$ [illegible] $n \times p$ [illegible] $\tilde{X}_{opt}$ [illegible] $r = 3200k^2/\beta\epsilon^2$ [illegible] EXACTLY(c) [illegible] 0.7

$$\left\| B - A\tilde{X}_{opt} \right\|_F \leq (1+\epsilon)\,\mathcal{Z}, \tag{21}$$

$$\left\| X_{opt} - \tilde{X}_{opt} \right\|_F \leq \frac{\epsilon}{\sigma_{\min}(A_k)}\mathcal{Z}. \tag{22}$$

[illegible] $\|U_{A,k}U_{A,k}^TB\|_F \geq \gamma \|B\|_F$ [illegible] $\gamma \in (0,1]$ [illegible] 0.7

$$\left\| X_{opt} - \tilde{X}_{opt} \right\|_F \leq \epsilon \left(\kappa(A_k)\sqrt{\gamma^{-2} - 1}\right) \|X_{opt}\|_F. \tag{23}$$

[illegible] $r = O(k \log k/\beta\epsilon^2)$ [illegible] EXPECTED(c) [illegible] 0.7 (21), (22) [illegible] (23) [illegible]

Equation (21) states that if the matrix of minimum-length vectors achieving the minimum in the sampled problem is substituted back into the residual norm for the original problem, then a good approximation to the original ℓ_2 regression problem is obtained. Equation (22) provides a bound for $||X_{opt} - \tilde{X}_{opt}||_F$ in terms of $\sigma_{\min}(A_k)$ and

$\mathcal{Z}$. If most of the "weight" of B lies in the complement of the column space of $A = A_k$ then this will provide a very poor approximation in terms of $\|X_{opt}\|_F$. However, if we also assume that a constant fraction of the "weight" of B lies in the subspace spanned by the columns of A, then we obtain the relative-error approximation of (23). Thus, Theorem 5 returns a good bound for $||X_{opt} - \tilde{X}_{opt}||_F$ if A_k is well-conditioned and if B lies "reasonably well" in the column space of A. Note that if the matrix of target vectors B lies completely within the column space of A, then $\mathcal{Z} = 0$ and $\gamma = 1$. In this case, Theorem 5 shows that Algorithm 3 returns $\tilde{\mathcal{Z}}$ and $\tilde{x}_{opt}$ that are exact solutions of the original ℓ_2 regression problem, independent of $\kappa(A_k)$. Finally, note that in our analysis of CX and [illegible] matrix decompositions we use only the result (21) from Theorem 5, but (22) and (23) are included for completeness.

6.2. Discussion of the method of "Subspace Sampling". An important aspect of Algorithm 3 is the nonuniform sampling probabilities (20) used by the Exactly(c) algorithm and the Expected(c) algorithm in the construction of the induced subproblem. We call sampling probabilities satisfying condition (20) [illegible]" probabilities. Condition (20) states that the sampling probabilities should be close to, or rather not much less than, the lengths, i.e., the Euclidean norms, of the rows of the left singular vectors of the matrix $A = A_k$. (Recall that in this section A is an $m \times n$ matrix with rank no more than k, and thus $U_{A,k}$ is an $m \times k$ matrix. Thus, the Euclidean norm of every [illegible] of $U_{A,k}$ equals 1, but the Euclidean norm of every [illegible] of $U_{A,k}$ is in general not equal and is only bounded above by 1.) Sampling probabilities of the form (20) should be contrasted with sampling probabilities that depend on the Euclidean norms of the columns or rows of A and that have received much attention recently [33, 34, 21, 22, 23, 26]. Since $A = U_A \Sigma_A V_A^T$, sampling probabilities with this latter form depend in a complicated manner on a mixture of subspace information (as found in U_A and V_A) and "size-of-A" information (as found in Σ_A). This convolution of information may account for their ability to capture coarse statistics such as approximating matrix multiplication or computing low-rank matrix approximations to additive error, but it also accounts for their difficulty in dealing with problems such as ℓ_2 regression or computing low-rank matrix approximations to relative error.

Since the solution of the ℓ_2 regression problem involves the computation of a pseudoinverse, the problem is not well-conditioned with respect to a perturbation (such as that introduced by sampling) that entails a change in dimensionality, even if (actually, especially if) that change in dimensionality corresponds to a small singular value. Since sampling probabilities satisfying (20) allow us to disentangle subspace information and "size-of-A" information, we will see that they will allow us to capture (with high probability) the [illegible] subspace of interest by sampling. More precisely, as we will see in Lemma 1, by using sampling probabilities that satisfy condition (20) and by choosing r appropriately, it will follow that

$$\text{rank}(DS^T U_{A,k}) = \text{rank}(U_{A,k}) = k.$$

Thus, the lengths of the Euclidean norms of the rows of $U_{A,k}$ may be interpreted as capturing a notion of information dispersal by the matrix A since they indicate to which part of the m-dimensional vector space the singular value information of A is being dispersed. In this case, condition (20) ensures that the sampling probabilities provide a bias toward the part of the high-dimensional constraint space to which A disperses its singular value information. Then, having constructed the sample, we will go to the low-dimensional, i.e., the r-dimensional rather than the m-dimensional

space, and approximate the ℓ_2 regression problem by doing computations that involve "size-of-A" information on the random sample.

This method of "subspace sampling" was first used in a preliminary version of the ℓ_2 regression results of this section [29]. Note that an immediate generalization of the results of [29] to the generalized ℓ_2 regression problem considered in this section would involve sampling probabilities of the form

(24)
$$p_i = \frac{(1/3)\left|(U_{A,k})_{(i)}\right|_2^2}{\sum_{j=1}^n \left|(U_{A,k})_{(j)}\right|_2^2} + \frac{(1/3)\left|(U_{A,k})_{(i)}\right|_2 \left(U_{A,k}^{\perp} U_{A,k}^{\perp}{}^T B\right)_i}{\sum_{j=1}^n \left|(U_{A,k})_{(j)}\right|_2 \left(U_{A,k}^{\perp} U_{A,k}^{\perp}{}^T B\right)_j} + \frac{(1/3)\left(U_{A,k}^{\perp} U_{A,k}^{\perp}{}^T B\right)_i^2}{\sum_{j=1}^n \left(U_{A,k}^{\perp} U_{A,k}^{\perp}{}^T B\right)_j^2},$$

rather than of the form (20). Since the second and third terms in (24) provide a bias toward the part of the complement of the column space of $A = A_k$ where B has significant weight, we directly obtain variance reduction. Thus, by using probabilities of the form (24) we can sample $O(k^2 \log(1/\delta)/\epsilon^2)$ columns and directly obtain the claims of Theorem 5 with probability at least $1 - \delta$. Although sampling probabilities of the form (20) are substantially simpler, we obtain variance control indirectly. We first establish that each of the claims of Theorem 5 holds with constant probability, and we then can show that each of the claims holds with probability at least $1 - \delta$ by running $O(\log(1/\delta))$ trials and using standard boosting procedures.

6.3. Proof of Theorem 5. In this section we provide a proof of Theorem 5. We will first prove (21), (22), and (23) under the assumption that the rows of A and B are sampled with the EXACTLY(c) algorithm. Then, in section 6.3.5, we will outline modifications to the proof if the rows of A and B are sampled with the EXPECTED(c) algorithm. For simplicity of notation in this section, we will let $\mathcal{S} = DS^T$ denote the $r \times m$ rescaled row-sampling matrix. Let the rank of the $m \times n$ matrix A be $\rho \leq k$, and let its SVD be

$$A = U_A \Sigma_A V_A^T,$$

where $U_A \in \mathbb{R}^{n \times \rho}$, $\Sigma_A \in \mathbb{R}^{\rho \times \rho}$, and $V_A \in \mathbb{R}^{d \times \rho}$. In addition, let the rank of the $r \times \rho$ matrix $\mathcal{S}U_A = DS^T U_A$ be $\tilde{\rho}$, and let its SVD be

$$\mathcal{S}U_A = U_{\mathcal{S}U_A} \Sigma_{\mathcal{S}U_A} V_{\mathcal{S}U_A}^T,$$

where $U_{\mathcal{S}U_A} \in \mathbb{R}^{r \times \tilde{\rho}}$, $\Sigma_{\mathcal{S}U_A} \in \mathbb{R}^{\tilde{\rho} \times \tilde{\rho}}$, and $V_{\mathcal{S}U_A} \in \mathbb{R}^{\rho \times \tilde{\rho}}$. Recall that $\tilde{\rho} \leq \rho \leq k \leq r$.

In order to illustrate the essential difficulty in constructing a sampling algorithm to approximate the solution of the generalized ℓ_2 regression problem, consider inserting $\tilde{X}_{opt} = (\mathcal{S}A_k)^+ \mathcal{S}B$ into $B - A_k X$:

$$\begin{aligned}
B - A_k \tilde{X}_{opt} &= B - A_k \left(\mathcal{S}A_k\right)^+ \mathcal{S}B \\
&= B - U_{A,k}\Sigma_{A,k}V_{A,k}^T \left(\mathcal{S}U_{A,k}\Sigma_{A,k}V_{A,k}^T\right)^+ \mathcal{S}B \\
&= B - U_{A,k}\Sigma_{A,k} \left(\mathcal{S}U_{A,k}\Sigma_{A,k}\right)^+ \mathcal{S}B \\
&= B - U_{A,k}\Sigma_{A,k} \left(U_{\mathcal{S}U_{A,k}}\Sigma_{\mathcal{S}U_{A,k}}V_{\mathcal{S}U_{A,k}}^T\Sigma_{A,k}\right)^+ \mathcal{S}B \\
&= B - U_{A,k}\Sigma_{A,k} \left(\Sigma_{\mathcal{S}U_{A,k}}V_{\mathcal{S}U_{A,k}}^T\Sigma_{A,k}\right)^+ U_{\mathcal{S}U_{A,k}}^T \mathcal{S}B.
\end{aligned}$$

To proceed further, we must deal with the pseudoinverse, which is not well-behaved with respect to perturbations that involve a change in dimensionality. To deal with this, we will focus on probabilities that depend on the subspace that we are down-

sampling, i.e., that depend on $U_{A,k}$, in order to guarantee that we capture the full subspace of interest.

6.3.1. Several lemmas of general interest. In this subsection, we will present three lemmas of general interest. Then, in the next subsections, we will use these lemmas to prove each of the claims of Theorem 5.

Since the $m \times k$ matrix $U_{A,k}$ is a matrix with orthogonal columns, several properties hold for it. For example, $\text{rank}(U_{A,k}) = k$, $U^+ = U^T$, and $A_k^+ = V_{A,k}\Sigma_{A,k}^{-1}U_{A,k}^T$. Although the $r \times k$ matrix $\mathcal{S}U_{A,k}$ does not have orthogonal columns, the following lemma characterizes the manner in which each of these three properties holds, either exactly or approximately. For the first lemma, r depends quadratically on k.

LEMMA 1. [illegible] $\epsilon \in (0,1]$ [illegible] $\Omega = (\mathcal{S}U_{A,k})^+ - (\mathcal{S}U_{A,k})^T$ [illegible] [illegible] (20) [illegible] $r \geq 400k^2/\beta\epsilon^2$ [illegible] [illegible] 0.9

$$\tilde{\rho} = \rho \text{ [illegible] } (\mathcal{S}U_{A,k}) = \text{ [illegible] } (U_{A,k}) = \text{ [illegible] } (A_k), \tag{25}$$

$$\|\Omega\|_2 = \left\|\Sigma_{\mathcal{S}U_{A,k}}^{-1} - \Sigma_{\mathcal{S}U_{A,k}}\right\|_2, \tag{26}$$

$$(\mathcal{S}A_k)^+ = V_{A,k}\Sigma_{A,k}^{-1}\left(\mathcal{S}U_{A,k}\right)^+, \tag{27}$$

$$\left\|\Sigma_{\mathcal{S}U_{A,k}} - \Sigma_{\mathcal{S}U_{A,k}}^{-1}\right\|_2 \leq \epsilon/\sqrt{2}. \tag{28}$$

[illegible]. To prove the first claim, note that for all $i \in [\rho]$

$$\begin{aligned}\left|1 - \sigma_i^2\left(\mathcal{S}U_{A,k}\right)\right| &= \left|\sigma_i\left(U_{A,k}^T U_{A,k}\right) - \sigma_i\left(U_{A,k}^T \mathcal{S}^T \mathcal{S}U_{A,k}\right)\right| \\ &\leq \left\|U_{A,k}^T U_{A,k} - U_{A,k}^T \mathcal{S}^T \mathcal{S}U_{A,k}\right\|_2 \qquad (29) \\ &\leq \left\|U_{A,k}^T U_{A,k} - U_{A,k}^T \mathcal{S}^T \mathcal{S}U_{A,k}\right\|_F. \qquad (30)\end{aligned}$$

Note that (29) follows from Corollary 8.1.6 of [37], and (30) follows since $\|\cdot\|_2 \leq \|\cdot\|_F$. To bound the error of approximating $U_{A,k}^T U_{A,k}$ by $U_{A,k}^T \mathcal{S}^T \mathcal{S}U_{A,k}$, we apply Theorem 6 of Appendix A. Since the sampling probabilities p_i satisfy (20), it follows from Theorem 6 and by applying Markov's inequality that with probability at least 0.9:

$$\begin{aligned}\left\|U_{A,k}^T U_{A,k} - U_{A,k}^T \mathcal{S}^T \mathcal{S}U_{A,k}\right\|_F &\leq 10\,\mathbf{E}\left[\left\|U_{A,k}^T U_{A,k} - U_{A,k}^T \mathcal{S}^T \mathcal{S}U_{A,k}\right\|_F\right] \\ &\leq \frac{10}{\sqrt{\beta r}}\left\|U_{A,k}\right\|_F^2, \qquad (31)\end{aligned}$$

where $\mathbf{E}[\cdot]$ denotes the expectation operator. By combining (30) and (31), recalling that $\|U_{A,k}\|_F^2 = \rho \leq k$, and using the assumed choice of r, it follows that

$$\left|1 - \sigma_i^2\left(\mathcal{S}U_{A,k}\right)\right| \leq \epsilon/2 \leq 1/2$$

since $\epsilon \leq 1$. This implies that all singular values of $\mathcal{S}U_{A,k}$ are strictly positive, and thus that $\text{rank}(\mathcal{S}U_{A,k}) = \text{rank}(U_{A,k}) = \text{rank}(A_k)$, which establishes the first claim.

To prove the second claim, we use the SVD of $\mathcal{S}U_{A,k}$ and note that

$$\begin{aligned}\|\Omega\|_2 &= \left\|\left(\mathcal{S}U_{A,k}\right)^+ - \left(\mathcal{S}U_{A,k}\right)^T\right\|_2 \\ &= \left\|\left(U_{\mathcal{S}U_{A,k}}\Sigma_{\mathcal{S}U_{A,k}}V_{\mathcal{S}U_{A,k}}^T\right)^+ - \left(U_{\mathcal{S}U_{A,k}}\Sigma_{\mathcal{S}U_{A,k}}V_{\mathcal{S}U_{A,k}}^T\right)^T\right\|_2 \\ &= \left\|V_{\mathcal{S}U_{A,k}}\left(\Sigma_{\mathcal{S}U_{A,k}}^{-1} - \Sigma_{\mathcal{S}U_{A,k}}\right)U_{\mathcal{S}U_{A,k}}^T\right\|_2.\end{aligned}$$

The claim follows since $V_{\mathcal{S}U_{A,k}}$ and $U_{\mathcal{S}U_{A,k}}$ are matrices with orthonormal columns.

To prove the third claim, note that

$$\begin{aligned}(\mathcal{S}A_k)^+ &= \left(\mathcal{S}U_{A,k}\Sigma_{A,k}V_{A,k}^T\right)^+ \\ &= \left(U_{\mathcal{S}U_{A,k}}\Sigma_{\mathcal{S}U_{A,k}}V_{\mathcal{S}U_{A,k}}^T\Sigma_{A,k}V_{A,k}^T\right)^+ \\ &= V_{A,k}\left(\Sigma_{\mathcal{S}U_{A,k}}V_{\mathcal{S}U_{A,k}}^T\Sigma_{A,k}\right)^+ U_{\mathcal{S}U_{A,k}}^T. \qquad (32)\end{aligned}$$

To remove the pseudoinverse in the above derivations, notice that since $\rho = \tilde{\rho}$ with probability at least 0.9, all three matrices $\Sigma_{\mathcal{S}U_{A,k}}$, $V_{\mathcal{S}U_{A,k}}$, and $\Sigma_{A,k}$ are full rank square $\rho \times \rho$ matrices, and thus are invertible. In this case,

$$\begin{aligned}\left(\Sigma_{\mathcal{S}U_{A,k}}V_{\mathcal{S}U_{A,k}}^T\Sigma_{A,k}\right)^+ &= \left(\Sigma_{\mathcal{S}U_{A,k}}V_{\mathcal{S}U_{A,k}}^T\Sigma_{A,k}\right)^{-1} \\ &= \Sigma_{A,k}^{-1}V_{\mathcal{S}U_{A,k}}\Sigma_{\mathcal{S}U_{A,k}}^{-1}. \qquad (33)\end{aligned}$$

By combining (32) and (33) we have that

$$\begin{aligned}(\mathcal{S}A_k)^+ &= V_{A,k}\Sigma_{A,k}^{-1}V_{\mathcal{S}U_{A,k}}\Sigma_{\mathcal{S}U_{A,k}}^{-1}U_{\mathcal{S}U_{A,k}}^T \\ &= V_{A,k}\Sigma_{A,k}^{-1}\left(\mathcal{S}U_{A,k}\right)^+,\end{aligned}$$

which establishes the third claim.[2]

Finally, to prove the fourth claim, recall that under the assumptions of the lemma $\rho = \tilde{\rho}$ with probability at least 0.9, and thus $\sigma_i\left(\mathcal{S}U_{A,k}\right) > 0$ for all $i \in [\rho]$. Thus,

$$\begin{aligned}\left\|\Sigma_{\mathcal{S}U_{A,k}}^{-1} - \Sigma_{\mathcal{S}U_{A,k}}\right\|_2 &= \max_{i,j\in[\rho]}\left|\sigma_i\left(\mathcal{S}U_{A,k}\right) - \frac{1}{\sigma_j\left(\mathcal{S}U_{A,k}\right)}\right| \\ &= \max_{i,j\in[\rho]}\frac{\left|\sigma_i\left(\mathcal{S}U_{A,k}\right)\sigma_j\left(\mathcal{S}U_{A,k}\right) - 1\right|}{\left|\sigma_j\left(\mathcal{S}U_{A,k}\right)\right|} \\ &\leq \max_{j\in[\rho]}\frac{\left|\sigma_j^2\left(\mathcal{S}U_{A,k}\right) - 1\right|}{\left|\sigma_j\left(\mathcal{S}U_{A,k}\right)\right|}. \qquad (34)\end{aligned}$$

Using the fact that, by (29), for all $i \in [\rho]$,

$$\left|1 - \sigma_i^2\left(\mathcal{S}U_{A,k}\right)\right| \leq \left\|U_{A,k}^TU_{A,k} - U_{A,k}^T\mathcal{S}^T\mathcal{S}U_{A,k}\right\|_2,$$

it follows that for all $i \in [\rho]$

$$\frac{1}{\sigma_i\left(\mathcal{S}U_{A,k}\right)} \leq \frac{1}{\sqrt{1 - \left\|U_{A,k}^TU_{A,k} - U_{A,k}^T\mathcal{S}^T\mathcal{S}U_{A,k}\right\|_2}}.$$

When these are combined with (34) it follows that

$$\left\|\Sigma_{\mathcal{S}U_{A,k}} - \Sigma_{\mathcal{S}U_{A,k}}^{-1}\right\|_2 \leq \frac{\left\|U_{A,k}^TU_{A,k} - U_{A,k}^T\mathcal{S}^T\mathcal{S}U_{A,k}\right\|_2}{\sqrt{1 - \left\|U_{A,k}^TU_{A,k} - U_{A,k}^T\mathcal{S}^T\mathcal{S}U_{A,k}\right\|_2}}.$$

[2] One might be tempted to suggest that the proof of this third claim should be "simplified" by appealing to the result that the generalized inverse of the product of two matrices equals the product of the generalized inverse of those matrices. This result is, of course, false—see, e.g., section 3.1.1 of [60]—and so we need a more refined analysis such as the one presented here.

Combining this with the Frobenius norm bound of (31), and noticing that our choice for r guarantees that $1 - ||U_{A,k}^T U_{A,k} - U_{A,k}^T \mathcal{S}^T \mathcal{S} U_{A,k}||_2 \geq 1/2$, concludes the proof of the fourth claim.

This concludes the proof of the lemma. □

The next lemma provides an approximate matrix multiplication bound that is useful in the proof of Theorem 5. For this lemma, r depends linearly on k.

LEMMA 2. [illegible] $\epsilon \in (0,1]$ [illegible] (20) [illegible] $r \geq 400k/\beta\epsilon^2$ [illegible] 0.9

$$\left\| U_{A,k}^T \mathcal{S}^T \mathcal{S} U_{A,k}^\perp {U_{A,k}^\perp}^T B \right\|_F \leq \frac{\epsilon}{2} \left\| U_{A,k}^\perp {U_{A,k}^\perp}^T B \right\|_F .$$

[illegible]. First, note that since $U_{A,k}$ is an orthogonal matrix and since $U_{A,k}^T U_{A,k}^\perp = 0$, we have that

$$\begin{aligned} \left\| U_{A,k}^T \mathcal{S}^T \mathcal{S} U_{A,k}^\perp {U_{A,k}^\perp}^T B \right\|_F &= \left\| U_{A,k} U_{A,k}^T \mathcal{S}^T \mathcal{S} U_{A,k}^\perp {U_{A,k}^\perp}^T B \right\|_F \\ &= \left\| U_{A,k} U_{A,k}^T U_{A,k}^\perp {U_{A,k}^\perp}^T B - U_{A,k} U_{A,k}^T \mathcal{S}^T \mathcal{S} U_{A,k}^\perp {U_{A,k}^\perp}^T B \right\|_F . \end{aligned} \tag{35}$$

Since $|(U_{A,k} U_{A,k}^T)_{(i)}|_2 = |(U_{A,k}^T)_{(i)}|_2$, the sampling probabilities (20) satisfy (45), where (45) will appear in Appendix A.2, and thus are appropriate for bounding the right-hand side of (35). Thus, it follows from Markov's inequality and Theorem 6 that with probability at least 0.9:

$$\begin{aligned} \left\| U_{A,k}^T \mathcal{S}^T \mathcal{S} U_{A,k}^\perp {U_{A,k}^\perp}^T B \right\|_F &\leq 10\, \mathbf{E} \left[\left\| U_{A,k}^T \mathcal{S}^T \mathcal{S} U_{A,k}^\perp {U_{A,k}^\perp}^T B \right\|_F \right] \\ &\leq \frac{10}{\sqrt{\beta r}} \left\| U_{A,k} U_{A,k}^T \right\|_F \left\| U_{A,k}^\perp {U_{A,k}^\perp}^T B \right\|_F . \end{aligned}$$

The lemma follows by the choice of r and since $||U_{A,k} U_{A,k}^T||_F = \sqrt{\rho} \leq \sqrt{k}$. □

The final lemma of this subsection relates the norm of the $m \times p$ matrix $U_{A,k}^\perp {U_{A,k}^\perp}^T B$ to the norm of the $r \times p$ matrix $\mathcal{S} U_{A,k}^\perp {U_{A,k}^\perp}^T B$, i.e., the row sampled and rescaled version of the original $m \times p$ matrix. For this lemma, r is independent of k.

LEMMA 3. [illegible] 0.9

$$\left\| \mathcal{S} U_{A,k}^\perp {U_{A,k}^\perp}^T B \right\|_F \leq 10 \left\| U_{A,k}^\perp {U_{A,k}^\perp}^T B \right\|_F .$$

[illegible]. Let $Q = U_{A,k}^\perp {U_{A,k}^\perp}^T B$, and let $j_1, j_2, \ldots, j_r$ be the r rows of Q that were included in $\mathcal{S}Q = DS^T Q$. Clearly,

$$\mathbf{E} \left[\left\| DS^T Q \right\|_F^2 \right] = \mathbf{E} \left[\sum_{t=1}^r \left| Q_{(j_t)} \right|_2^2 \right] = \sum_{t=1}^r \mathbf{E} \left[\left| Q_{(j_t)} \right|_2^2 \right] = \sum_{t=1}^r \sum_{j=1}^n p_j \frac{\left| Q_{(j)} \right|_2^2}{r p_j} = \left\| Q \right\|_F^2 , \tag{36}$$

where the penultimate equality follows by evaluating the expectation. The lemma follows by applying Markov's inequality and taking the square root of both sides of the resulting inequality. □

6.3.2. Proof of (21). In this subsection, we will bound $B - A_k \tilde{X}_{opt}$, thus proving (21). For the moment, let us assume that $r = 400k^2/\beta\epsilon^2$, in which case the assumption on r is satisfied for each of Lemma 1, Lemma 2, and Lemma 3. Thus, the claims of

all three lemmas hold simultaneously with probability at least $1 - 3(0.1) \geq 0.7$, and so let us condition on this event.

First, we have that

$$\begin{aligned} B - A_k\tilde{X}_{opt} &= B - A_k \left(\mathcal{S}A_k\right)^+ \mathcal{S}B \\ &= B - U_{A,k} \left(\mathcal{S}U_{A,k}\right)^+ \mathcal{S}B \qquad (37) \\ &= B - U_{A,k} \left(\mathcal{S}U_{A,k}\right)^+ \mathcal{S}U_{A,k}U_{A,k}^T B - U_{A,k} \left(\mathcal{S}U_{A,k}\right)^+ \mathcal{S}U_{A,k}^\perp {U_{A,k}^\perp}^T B \qquad (38) \\ &= U_{A,k}^\perp {U_{A,k}^\perp}^T B - U_{A,k} \left(\mathcal{S}U_{A,k}\right)^+ \mathcal{S}U_{A,k}^\perp {U_{A,k}^\perp}^T B. \qquad (39) \end{aligned}$$

Equation (37) follows from (27) of Lemma 1, (38) follows by inserting $U_{A,k}U_{A,k}^T + U_{A,k}^\perp {U_{A,k}^\perp}^T = I_n$, and (39) follows since $\left(\mathcal{S}U_{A,k}\right)^+ \mathcal{S}U_{A,k} = I_\rho$ by Lemma 1. We emphasize that $\left(\mathcal{S}U_{A,k}\right)^+ \mathcal{S}U_{A,k} = V_{\mathcal{S}U_{A,k}} V_{\mathcal{S}U_{A,k}}^T = I_\rho$ does not hold for general sampling methods, but it does hold in this case since $\tilde{\rho} = \rho$, which follows from Lemma 1.

By taking the Frobenius norm of both sides of (39), by using the triangle inequality, and recalling that $\Omega = \left(\mathcal{S}U_{A,k}\right)^+ - \left(\mathcal{S}U_{A,k}\right)^T$, we have that

$$\begin{aligned} \left\|B - A_k\tilde{X}_{opt}\right\|_F &\leq \left\|U_{A,k}^\perp {U_{A,k}^\perp}^T B\right\|_F + \left\|U_{A,k} \left(\mathcal{S}U_{A,k}\right)^T \mathcal{S}U_{A,k}^\perp {U_{A,k}^\perp}^T B\right\|_F \\ &\quad + \left\|U_{A,k}\Omega\mathcal{S}U_{A,k}^\perp {U_{A,k}^\perp}^T B\right\|_F \\ &\leq \left\|U_{A,k}^\perp {U_{A,k}^\perp}^T B\right\|_F + \left\|U_{A,k}^T \mathcal{S}^T \mathcal{S}U_{A,k}^\perp {U_{A,k}^\perp}^T B\right\|_F \qquad (40) \\ &\quad + \|\Omega\|_2 \left\|\mathcal{S}U_{A,k}^\perp {U_{A,k}^\perp}^T B\right\|_F, \end{aligned}$$

where (40) follows by submultiplicativity and since $U_{A,k}$ has orthogonal columns. By combining (40) with the bounds provided by Lemma 1 through Lemma 3, it follows that

$$\begin{aligned} \left\|B - A_k\tilde{X}_{opt}\right\|_F &\leq (1 + \epsilon/2 + 10\epsilon/\sqrt{2})\mathcal{Z} \\ &\leq (1 + 8\epsilon)\mathcal{Z}. \end{aligned}$$

Equation (21) follows by setting $\epsilon' = \epsilon/8$ and using the value of r assumed by the theorem.

6.3.3. Proof of (22). In this subsection, we will provide a bound for $||\tilde{X}_{opt} - X_{opt}||_F$ in terms of $\mathcal{Z}$, thus proving (22). For the moment, let us assume that $r = 400k^2/\beta\epsilon^2$, in which case the assumption on r is satisfied for each of Lemma 1, Lemma 2, and Lemma 3. Thus, the claims of all three lemmas hold simultaneously with probability at least $1 - 3(0.1) \geq 0.7$, and so let us condition on this event.

Since $U_{A,k}U_{A,k}^T + U_{A,k}^\perp {U_{A,k}^\perp}^T = I_n$ and $\left(\mathcal{S}U_{A,k}\right)^+ \mathcal{S}U_{A,k} = I_\rho$, we have that

$$\begin{aligned} X_{opt} - \tilde{X}_{opt} &= A_k^+ B - \left(\mathcal{S}A_k\right)^+ \mathcal{S}B \\ &= V_{A,k}\Sigma_{A,k}^{-1}U_{A,k}^T B - V_{A,k}\Sigma_{A,k}^{-1} \left(\mathcal{S}U_{A,k}\right)^+ \mathcal{S}B \\ &= V_{A,k}\Sigma_{A,k}^{-1}U_{A,k}^T B - V_{A,k}\Sigma_{A,k}^{-1} \left(\mathcal{S}U_{A,k}\right)^+ \mathcal{S}U_{A,k}U_{A,k}^T B \\ &\quad - V_{A,k}\Sigma_{A,k}^{-1} \left(\mathcal{S}U_{A,k}\right)^+ \mathcal{S}U_{A,k}^\perp {U_{A,k}^\perp}^T B \\ &= -V_{A,k}\Sigma_{A,k}^{-1} \left(\mathcal{S}U_{A,k}\right)^+ \mathcal{S}U_{A,k}^\perp {U_{A,k}^\perp}^T B. \end{aligned}$$

Thus, it follows that

$$\begin{aligned}
\left\|X_{opt} - \tilde{X}_{opt}\right\|_F &= \left\|V_{A,k}\Sigma_{A,k}^{-1}\left(\mathcal{S}U_{A,k}\right)^{+}\mathcal{S}U_{A,k}^{\perp}{U_{A,k}^{\perp}}^{T}B\right\|_F \\
&= \left\|\Sigma_{A,k}^{-1}\left(\left(\mathcal{S}U_{A,k}\right)^{T} + \Omega\right)\mathcal{S}U_{A,k}^{\perp}{U_{A,k}^{\perp}}^{T}B\right\|_F \\
&\leq \frac{1}{\sigma_{min}(A_k)}\left\|\left(\mathcal{S}U_{A,k}\right)^{T}\mathcal{S}U_{A,k}^{\perp}{U_{A,k}^{\perp}}^{T}B\right\|_F \\
&\quad + \frac{1}{\sigma_{min}(A_k)}\left\|\Omega\mathcal{S}U_{A,k}^{\perp}{U_{A,k}^{\perp}}^{T}B\right\|_F \\
&\leq \frac{1}{\sigma_{min}(A_k)}\left\|U_{A,k}^{T}\mathcal{S}^{T}\mathcal{S}U_{A,k}^{\perp}{U_{A,k}^{\perp}}^{T}B\right\|_F \\
&\quad + \frac{1}{\sigma_{min}(A_k)}\left\|\Omega\right\|_2\left\|\mathcal{S}U_{A,k}^{\perp}{U_{A,k}^{\perp}}^{T}B\right\|_F .
\end{aligned} \tag{41}$$

By combining (41) with Lemmas 1, 2, and 3, it follows that

$$\begin{aligned}
\left\|\tilde{X}_{opt} - X_{opt}\right\|_F &\leq \sigma_{\min}^{-1}(A_k)\left(\epsilon/2 + 10\epsilon/\sqrt{2}\right)\mathcal{Z} \\
&\leq \frac{8\epsilon}{\sigma_{\min}(A_k)}\mathcal{Z}.
\end{aligned}$$

Equation (22) follows by setting $\epsilon' = \epsilon/8$ and using the value of r assumed by the theorem.

6.3.4. Proof of (23). The error bound provided by (22) could be quite weak, since $\min_{X\in\mathbb{R}^{n\times p}}\left\|B - A_kX\right\|_F$ could be quite close or even equal to $\left\|B\right\|_F$, if B has most or all of its "weight" outside of the column space of A_k. Under a slightly stronger assumption, we will provide a bound $||\tilde{X}_{opt} - X_{opt}||_F$ in terms of $\left\|X_{opt}\right\|_F$, thus proving (23).

If we make the additional assumption that a [illegible] of the "weight" of B lies in the subspace spanned by the columns of A_k, then it follows that

$$\begin{aligned}
\mathcal{Z}^2 &= \left(\min_{X\in\mathbb{R}^{n\times p}}\left\|B - A_kX\right\|_F\right)^2 \\
&= \left\|U_{A,k}^{\perp}{U_{A,k}^{\perp}}^{T}B\right\|_F^2 \\
&= \left\|B\right\|_F^2 - \left\|U_{A,k}U_{A,k}^{T}B\right\|_F^2 \\
&\leq \left(\gamma^{-2} - 1\right)\left\|U_{A,k}U_{A,k}^{T}B\right\|_F^2 .
\end{aligned} \tag{42}$$

In order to relate $||U_{A,k}U_{A,k}^{T}B||_F$ and thus $\mathcal{Z}$ to $\left\|X_{opt}\right\|_F$ note that

$$\begin{aligned}
\left\|X_{opt}\right\|_F &= \left\|V_{A,k}\Sigma_{A,k}^{-1}U_{A,k}^{T}B\right\|_F \\
&= \left\|\Sigma_{A,k}^{-1}U_{A,k}^{T}B\right\|_F \\
&\geq \sigma_{\min}(\Sigma_{A,k}^{-1})\left\|U_{A,k}^{T}B\right\|_F \\
&= \frac{\left\|U_{A,k}U_{A,k}^{T}B\right\|_F}{\sigma_{\max}(A_k)}.
\end{aligned} \tag{43}$$

By combining (22) with (42) and (43), we get

$$\begin{aligned}\left\|\tilde{X}_{opt} - X_{opt}\right\|_F &\leq \frac{\epsilon}{\sigma_{\min}(A_k)}\mathcal{Z} \\ &\leq \frac{\epsilon}{\sigma_{\min}(A_k)}\sqrt{\gamma^{-2}-1}\left\|U_{A,k}U_{A,k}^T B\right\|_F \\ &\leq \epsilon\frac{\sigma_{\max}(A_k)}{\sigma_{\min}(A_k)}\sqrt{\gamma^{-2}-1}\left\|X_{opt}\right\|_F,\end{aligned}$$

which establishes (23).

6.3.5. Modifications to the proof with alternate row sampling procedure. If, in Algorithm 3, the rows are sampled with the EXPECTED(c) algorithm, then the proof of the claims of Theorem 5 is analogous to the proof described in the four previous subsections, with the following major exception. The claims of Lemma 1 hold if $r = O(k \log k/\beta\epsilon^2)$ rows are chosen with the EXPECTED(c) algorithm. To see this, recall that to bound the first claim of Lemma 1, we must bound the spectral norm $\|U_{A,k}^T U_{A,k} - U_{A,k}^T \mathcal{S}^T \mathcal{S} U_{A,k}\|_2$ in (29). If the sampling is performed with the EXACTLY(c) algorithm, then this is bounded in (30) by the corresponding Frobenius norm, which is then bounded with Theorem 6. On the other hand, if the sampling is performed with the EXPECTED(c) algorithm, then we can bound (29) directly with the spectral norm bound provided by Theorem 7.

Since the remaining claims of Lemma 1 follow from the first, they are also valid if $r = O(k \log k/\beta\epsilon^2)$ rows are chosen with the EXPECTED(c) algorithm. Lemma 2 still follows if $r = 400k/\beta\epsilon^2$, by using the Frobenius norm bound of Theorem 7, and Lemma 3 also follows immediately. The proofs of (21), (22), and (23) are identical, and thus Theorem 5, under the assumption that the the rows are chosen with the EXPECTED(c) algorithm, follows.

7. Empirical evaluation. Although this is a theoretical paper, it is motivated by applications, and thus one might wonder about the empirical applicability of our methods. For example, if we want to do as well as the best rank $k = 100$ (respectively, $k = 10$) approximation, with relative error bound $\epsilon = 0.1$, then our main theorem samples 3.2 billion (respectively, 32 million) columns of the matrix A using the EXACTLY(c) algorithm. Of course, our main theorem states that it in order to obtain our strong provable worst-case relative-error guarantees it [illegible] to choose that many columns. But, it would be a source of concern if anything like that number of columns is needed in "real" scientific and internet data applications.

In this section, we provide an empirical evaluation of the performance of our two main sampling procedures both for CX and [illegible] decompositions. In particular, we will evaluate how well the proposed column/row selection strategies perform at capturing the Frobenius norm for matrices derived from DNA SNP analysis, recommendation system analysis, and term-document analysis. By applying our algorithms to data sets drawn from these three diverse domains of modern data analysis, we will demonstrate that we can obtain very good Frobenius norm reconstruction by sampling a number of columns and/or rows that equals a small constant, e.g., 2 or 3 or 4 (as opposed to, e.g., a million or a billion), times the rank parameter k.

7.1. Details of our empirical evaluation. The empirical evaluation of our CX and [illegible] matrix decompositions has been performed using the following two types of column/row selection methods:

- "Subspace sampling" (with replacement) using the EXACTLY(c) algorithm; and
- "Subspace sampling" (without replacement) using the EXPECTED(c) algorithm.

In addition, the empirical evaluation has been performed on the following three data sets:

- Matrices derived from the DNA SNP HapMap data [15, 52]—see section 7.2;
- A matrix derived from the Jester recommendation system corpus [35, 48]—see section 7.3; and
- A matrix derived from the Reuters term-document corpus [46, 16]—see section 7.4.

We have chosen these three data sets on which to evaluate the empirical applicability of our algorithms for three reasons: first, these three application domains are representative of a wide range of areas of modern scientific and internet data analysis; second, these matrices are all approximately (to a greater or lesser extent) low-rank, and they are all data for which spectral methods such as low-rank approximations have been successfully applied; and third, we have already (with collaboraters from these application areas) applied our algorithms to these data sets [48, 52, 16]. In these data application papers [48, 52, 16], we have shown that our main CX and CUR decomposition algorithms (either the algorithms for which we have provable performance guarantees and/or greedy variants of these basic algorithms) perform well on tasks such as classification, denoising, reconstruction, prediction, and clustering—tasks that are of more immediate interest to data practitioners than simply capturing the norm of the data matrix.

In this section, however, we we will restrict ourselves to an empirical evaluation of our two main theorems. To do so, we will fix a rank parameter k, and we will present plots of the Frobenius norm error (normalized by $||A - A_k||_F$), as a function of the number of samples chosen. For example, we will consider $\Theta_1 \equiv ||A - CC^+A||_F / ||A - A_k||_F$, where A_k is the best rank-k approximation to the matrix A, as a function of the number c of columns chosen. This ratio corresponds to the quantity that is bounded by $1 + \epsilon$ in Theorem 3. For $c = k$, this quantity will be no less than 1; of course, if we choose $c > k$ columns, then this ratio may be less than 1. Following the remark after Theorem 3, we will also consider $\Theta_2 \equiv ||A - CC^+A_k||_F / ||A - A_k||_F$. This ensures that the approximation has a rank no greater than k (which is of interest in certain applications), and thus the plotted ratio will clearly be no less than 1, for every value of c. We will also consider $\Theta_3 \equiv ||A - CUR||_F / ||A - A_k||_F$, which corresponds to the quantity that is bounded by $1 + \epsilon$ in Theorem 4.

Two technical points should be noted about these plots in the upcoming subsections. First, we ran our CX or CUR decomposition algorithm several—e.g., three or five, depending on the size of the data being plotted—times (corresponding, say, to multiple runs to boost the δ failure probability), and the minimum value over these repetitions was returned; this was repeated several times, and the average of those values is plotted. Second, for the plots of $||A - CUR||_F / ||A - A_k||_F$, the number of rows selected is set to be twice the corresponding number of columns selected; optimizing over this would lead to marginally better performance than that presented.

7.2. DNA SNP HapMap data. Our first dataset comes from the field of human genetics. The HapMap project, a continuation of the Human Genome project, aims to map the loci in the human genome that differ among individuals [15]. The HapMap project focuses on the so-called SNPs, which are a very common type of

variation in the genome (nearly 10^7 such loci have been identified in the human genome). Significant motivation exists in the genetics community for minimizing the number of SNPs that must be assayed, and in [52], we demonstrated how , -type methods may be used to efficiently reconstruct unassayed SNPs from a small number of assayed SNPs.

Both in [52] and here, we consider two regions of the genome known as HOXB and 17q25. Three populations were studied for each region: Yoruban, a sub-Saharan African population; a European population; and a joint Japanese/Chinese population. Each population had 90 individuals, each corresponding to a row of the input matrix. Columns of each matrix correspond to SNPs within the HOXB or 17q25 regions. The genotypic data were encoded appropriately in order to be converted to numeric data in the form of matrices. (Careful preprocessing was done to remove fixed SNPs, as well as SNPs with too many missing entries, etc.) The HapMap project provided data on 370 SNPs in 17q25 and 571 SNP in HOXB [15]. Thus, for example, our matrix for the Yoruban population in HOXB is a 90×571 matrix, whose entries are in the set $\{-1, 0, +1\}$.[3] The other data matrices are of similar (not extremely large) size. See [52] and references therein for details.

Data not presented indicate that for all three populations and for both genomic regions, the data possess a great deal of linear structure. For example, in the 17q25 matrices, one needs 9, 9, and 7 singular vectors to capture 80% of the Frobenius norm for the Yoruban, European, and the Japanese/Chinese populations, respectively; and one needs 18, 16, and 13 singular vectors, respectively, to capture 90%. The matrices for the HOXB region of the genome are even more redundant; one needs only 7, 6, and 4 singular vectors, respectively, to capture 80% of the Frobenius norm.

In Figure 1, data are presented for the Yoruban HOXB data matrix. Each of the six subfigures presents a plot of the Frobenius norm error as a function of the number c of samples chosen. In particular, for two values of the rank parameter, i.e., $k = 5$ and $k = 10$, the ratio $\Theta_i = ||A - A'||_F / ||A - A_k||_F$ is plotted, where $A' = CC^+A$ for $i = 1$; $A' = CC^+A_k$ for $i = 2$; and $A' = CUR$ for $i = 3$. Clearly, in all these cases, only modest oversampling is needed to capture "nearly all" of the dominant part of the spectrum of the data matrix. For example, for $k = 5$: if $c = 5$, then $\Theta_1 = 1.12$; if $c \geq 6$, then $\Theta_1 < 1.1$; and if $c \geq 9$, then $\Theta_1 < 1.0$. Similarly, for $k = 10$: if $c = 10$, then $\Theta_1 = 1.22$; if $c \geq 15$, then $\Theta_1 < 1.1$; and if $c \geq 18$, then $\Theta_1 < 1.0$. Similar results hold if the projection onto the span of the columns is regularized through a rank-k space and also if rows are chosen after the columns. For example, for $k = 10$: if $c \geq 16$, then $\Theta_2 < 1.2$ and if $c \gtrsim 30$, then $\Theta_2 < 1.1$. Similarly, even though the computations for Θ_3 are slightly worse and somewhat noisier due to the second level of sampling (columns and then rows), the results still show that only modest oversampling (of c relative to k) is needed. For example, if $k = 10$, then $\Theta_3 < 1.1$ if $c \geq 20$ or $c \geq 28$, depending on precisely how the columns are chosen. Interestingly, in this last case, not only are the plots noisier, but the EXPECTED(c) algorithm and the EXACTLY(c) algorithm seem to lead to (slightly) different results as a function of c.

Qualitatively similar results are seen for the other populations and the other genomic regions. For example, in Figure 2, data are presented for the European

[3]The encoding should be interpreted as follows: each SNP consists of two alleles (nucleotide bases); these bases are the same for all humans. Say that these bases are A and G. Then a value of +1 corresponds to an individual whose genotype (pair of alleles) is AA, a value of 0 corresponds to an individual whose genotype is AG or GA, and a value of −1 corresponds to an individual whose genotype is GG.

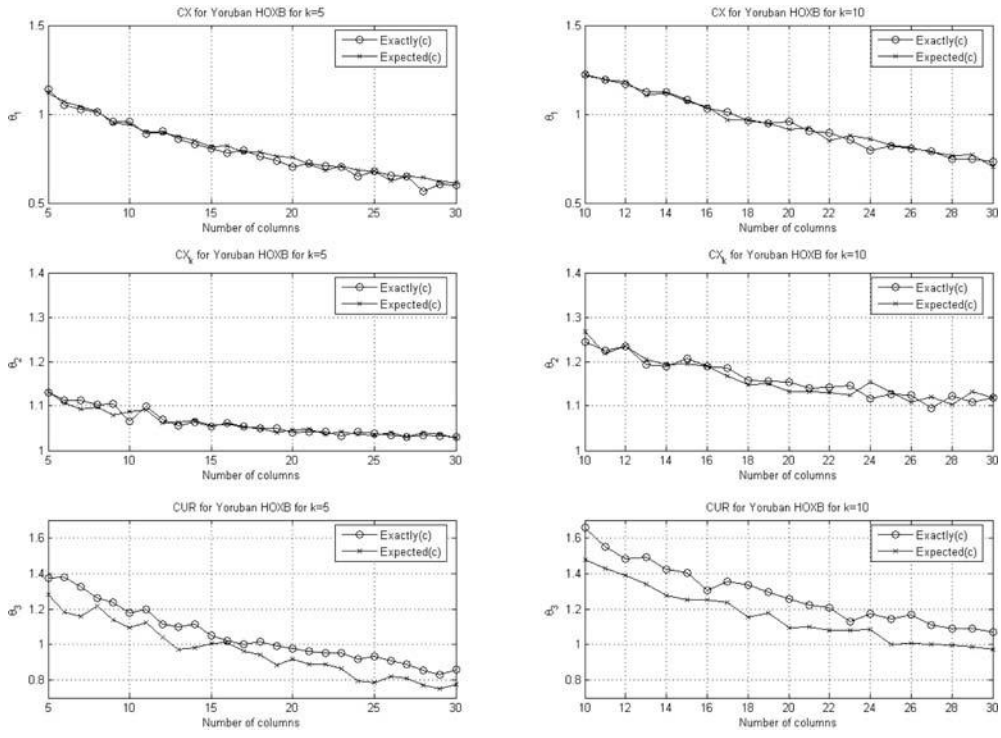


FIG. 1. *Reconstruction error for the Yoruban population in the HOXB region of the genome. Shown are* Θ_1, Θ_2, *and* Θ_3 *(as defined in the text) for two values of the rank parameter* k. *The X-axis corresponds to the number of columns sampled with the* EXACTLY(c) *algorithm or the* EXPECTED(c) *algorithm.*

population for both the HOXB and the 17q25 regions of the genome for the value of the rank parameter $k = 10$. For the HOXB region, $\Theta_1 = 1.36$ if $c = 10$ (this is higher than for the corresponding Yoruban data), $\Theta_1 < 1.0$ if $c \gtrsim 17$ (this is similar to the corresponding Yoruban data), and $\Theta_1 = 0.62$ if $c = 30$ (this is less than the Yoruban data). Similar trends are seen for Θ_2 and Θ_3 and also for the 17q25 region. In all cases, only very modest oversampling is needed for accurate Frobenius norm reconstruction. Data not presented indicate that the data for the joint Japanese/Chinese population is quite similar to or slightly better than those results presented.

7.3. Recommendation system jester data. Our second dataset comes from the field of recommendation system analysis, in which one is typically interested in making purchase recommendations to a user at an electronic commerce web site [35]. Collaborative methods (as opposed to content-based or hybrid) involve recommending to the user items that people with similar tastes or preferences liked in the past. Many collaborative filtering algorithms represent a user as an n dimensional vector, where n is the number of distinct products, and where the components of the vector are a measure of the rating provided by that user for that product. Thus, for a set of m users, the user-product ratings matrix is an $m \times n$ matrix A, where A_{ij} is the rating by user i for product j (or is null if the rating is not provided).

The so-called Jester joke dataset is a commonly used benchmark for recommendation system research and development [35]. In [48], we applied a , decomposition on this data to the problem of reconstructing missing entries and making accurate

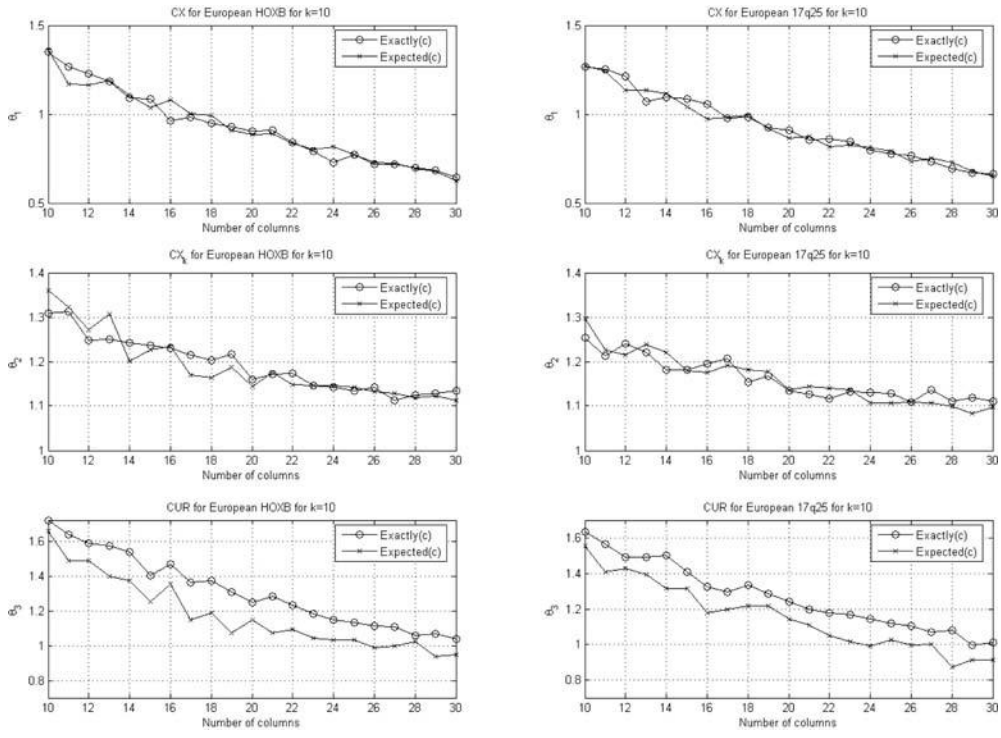


FIG. 2. *Reconstruction error for the European population in both the HOXB and 17q25 regions of the genome. Shown are* Θ_1, Θ_2, *and* Θ_3 *(as defined in the text) for* $k = 10$. *The X-axis corresponds to the number of columns sampled with the* EXACTLY(c) *algorithm or the* EXPECTED(c) *algorithm.*

recommendations. Here, we consider the $m = 14{,}116$ (out of ca. 73,000) users who rated all of the $n = 100$ products (i.e., jokes) in the Jester data. The entries in this $14{,}116 \times 100$ matrix A are real numbers between -10 and $+10$ that represent the user's rating of a product.

Figure 3 presents the empirical results for the Jester recommendation system data. The rank of the $14{,}116 \times 100$ matrix is 100, and although only seven singular vectors are needed to capture 50% of the Frobenius norm, 50 are needed to capture 80%, and 73 are needed to capture 90%. Thus, the spectrum and shape of this matrix (this matrix is very rectangular) are very different from that of the matrices of the previous subsection.

Figure 3 presents reconstruction error results for selecting columns (i.e., products or jokes), for selecting rows (i.e., users), and for selecting both columns and rows simultaneously. For example, when selecting columns from A, if $k = 15$, then $\Theta_1 = 1.14$ if $c = 15$, $\Theta_1 \leq 1$ if $c \gtrsim 29$, and $\Theta_1 = 0.99$ when $c = 30$. Although the matrix is very rectangular, quantitatively very similar results are obtained for the analogue of Θ_1 (called Θ_1^R in the figure) if rows are sampled (or, equivalently if columns are sampled from A^T). Thus, when our main CX decomposition is applied to either A or to A^T, a small number of columns (products) or rows (users), capture most of the Frobenius norm of A that is captured by the best rank k approximation to A. A similar result holds for the simultaneously choosing columns and rows of A (both users and products), and applying our CUR approximation algorithm. As with the data of the previous subsection, the data for Θ_3 are much noisier when both columns

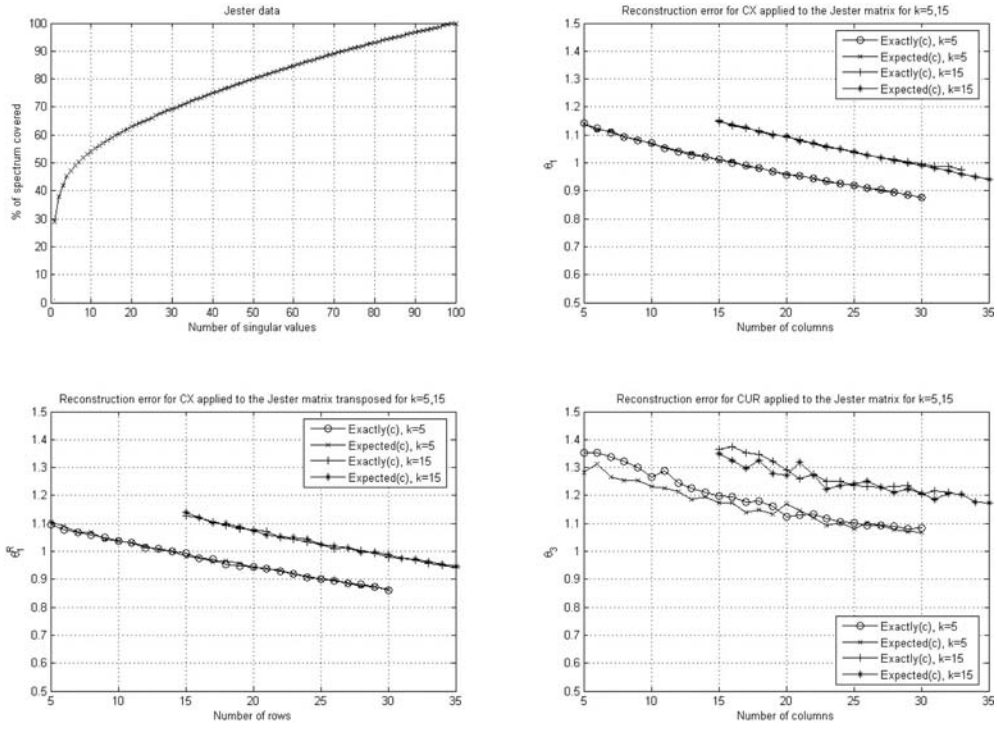


FIG. 3. *Empirical results for the Jester recommendation system data. Shown are: the percentage of the Frobenius norm captured as a function of the number of singular components;* Θ_1 *for sampling columns from* A *for* $k = 5$ *and* $k = 15$*; the analogue of* Θ_1 *for selecting rows from* A *(i.e.,* Θ_1 *for sampling columns from* A^T*); and* Θ_3 *for selecting columns and then rows for* $k = 5$ *and* $k = 15$.

and rows are chosen, but even in this case $\Theta_3 \leq 1.1$ for $k = 5$ if $c \gtrsim 25$ and $\Theta_3 \leq 1.2$ for $k = 15$ if $c \gtrsim 30$. In all these cases, data not presented indicate that qualitatively similar (but shifted) results are obtained for higher values of the rank parameter k.

7.4. Term-document Reuters data. Our third data set comes from the field of text categorization and information retrieval. In these applications, documents are often represented as a so-called "bag of words" and a vector space model is used. In 2000, Reuters Ltd made available a large collection of Reuters News stories for use in research and development of natural language processing, information retrieval, and machine learning systems. This corpus, known as "Reuters Corpus, Volume 1" or RCV1, is significantly larger (it contains over 800,000 news items from 1996-97) than the older, well-known Reuters-21578 collection, which has been heavily used in the text classification community [46]. In [16], we considered the problem of feature selection for improved classification, and we compared a CX-like column selection procedure to several traditional methods. The data come with class labels and possess a hierarchical class structure (which we used in [16]) which we ignored here. Here, we used the ,. -normalized term-document matrix and the training data from the one test-train split provided by Lewis, Yang, Rose, and Li [46]. Thus, the Reuters matrix we considered here is a (very sparse) 47,236 × 23,149 matrix whose elements are real numbers between 0 and 1 that represent a normalized frequency.

Figure 4 presents the empirical results for the Reuters term-document data. Note that this data is not only much larger than the data from the previous two subsec-

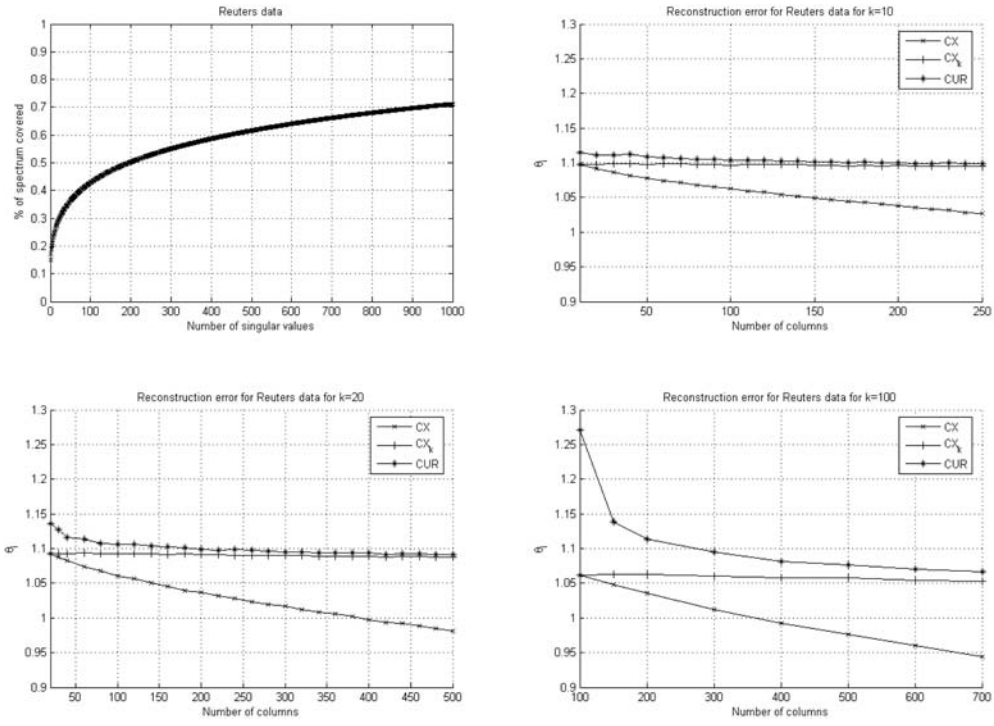


FIG. 4. *Empirical results for the Reuters term-document data. Shown are the percentage of the Frobenius norm captured as a function of the number of singular components;* Θ_1, Θ_2, *and* Θ_3 *as a function of the number of sampled columns and/or rows for three different values of the rank parameter* k.

tions, it is also less well-approximated by a low rank matrix. Less than 50% of the Frobenius norm is captured by the first $k = 100$ singular components, and less than 80% is captured by the first $k = 1500$ singular components. (Nevertheless, spectral methods have frequently been applied to this data.) The matrix is very sparse, and performing computations is expensive in terms of space and time (due to multiple randomized trials and since the dense matrices of singular vectors are large) if the rank parameter k is chosen to be more than a few hundred. Thus, to demonstrate the empirical applicability of our main algorithms, we considered several smaller values of k. Here, we report results for $k = 10$ and $c = 10$ to 250; for $k = 20$ and $c = 20$ to 500; and for $k = 100$ and $c = 100$ to 700. Note that we report results only for columns chosen with the EXPECTED(c) algorithm; initial unreported computations on several smaller systems indicate that very similar results will be obtained with the EXACTLY(c) algorithm.

In all of these cases, and for all values of Θ_1, Θ_2, and Θ_3, only modest oversampling leads to fairly small reconstruction error. The worst data point reported was for $\Theta_3 = 1.272$ for $k = 100$ and $c = 100$, and even in that case $\Theta_3 < 1.1$ for $c \geq 300$. Interestingly, all the curves tend to decrease somewhat more slowly (as a function of oversampling c, relative to k) than the corresponding curves in the previous subsections do. Note that Θ_1 does not decrease below 1.0 for $k = 10$ until after $c = 500$; for $k = 20$, it drops below 1.0 by $c = 400$, and for $k = 100$ (which obviously captures the largest fraction of the Frobenius norm) it drops below 1.0 at $c \approx 350$. Thus, this phenomenon is likely related to the degree to which the chosen value for

the rank parameter k captures a reasonable fraction of the norm of the original matrix. Nevertheless, in all cases, we can achieve $\Theta_i < 1.1$ with only a modest degree of oversampling c relative to k.

8. Conclusion. We have presented and analyzed randomized algorithms for computing low-rank matrix approximations that are explicitly expressed in terms of a small number of columns and/or rows of the input matrix. These algorithm achieve relative-error guarantees, whereas previous algorithms for these problems achieved only additive-error guarantees. These algorithms randomly sample in a novel manner that we call "subspace sampling," and their analysis amounts to approximating a generalized ℓ_2 regression problem by random sampling. As described in section 1.1 and in [52, 48], such low-rank matrix approximations have numerous applications for the improved analysis of data.

We conclude with several open problems:

- To what extent do the results of this paper generalize to other matrix norms?
- What hardness results can be established for the optimal choice of columns and/or rows?
- Does there exist a deterministic approximation algorithm for either of the problems we consider?
- Does there exist an efficient deterministic algorithm to choose columns and/or rows that exactly or approximately optimize the maximum volume of the induced parallelepiped? (As pointed out to us by an anonymous reviewer, [40, 61] provide such procedures; it would be interesting to see if bounds of the form we prove can be established for the algorithms of [40, 61]).
- Can we formulate a simple condition that we can check after we have sampled the columns and/or rows to determine whether we have achieved a $1 + \epsilon$ approximation with that sample?
- Can we obtain similar algorithms and comparable bounds for formulations of these problems that include regularization and/or conditioning?
- What heuristic variants of these algorithms are most appropriate in different application domains?
- Are the algorithms presented in this paper numerically stable?

Appendix A. Approximating matrix multiplication. In this section, we describe two complementary procedures for randomly sampling (and rescaling) columns and/or rows from an input matrix. Then, we describe an algorithm for approximating the product of two matrices by randomly sampling columns and rows from the input matrices using one of the two sampling procedures.

A.1. Sampling columns and rows from matrices. We describe two simple algorithms for randomly sampling a set of columns from an input matrix. Each algorithm takes as input an $m \times n$ matrix A and a probability distribution $\{p_i\}_{i=1}^n$, and each constructs a matrix C consisting of a rescaled copy of a small number of columns from A. Clearly, each algorithm can be modified to sample rows from a matrix. The first algorithm is the EXACTLY(c) algorithm, which is described in Algorithm 4 using the sampling matrix formalism described in section 2. In this algorithm, c columns [illegible] of A are chosen in c i.i.d. trials, where in each trial the ith column of A is picked with probability p_i. Note that because the sampling is performed with replacement, a single column of A may be included in C more than once. The second algorithm is the EXPECTED(c) algorithm, which is described in Algorithm 5, also using the sampling matrix formalism described in section 2. In this algorithm, at most c columns [illegible] of A are chosen by including the

ith column of A in C with probability $\tilde{p}_i = \min\{1, cp_i\}$. Note that the exact value of the number of columns returned is not known before the execution of this second algorithm; we do not perform an analysis of this random variable.

Data : $A \in \mathbb{R}^{m\times n}$, $p_i \geq 0, i \in [n]$ s.t. $\sum_{i\in[n]} p_i = 1$, positive integer $c \leq n$.
Result : Sampling matrix S, rescaling matrix D, and sampled and rescaled columns C.
Initialize S and D to the all zeros matrices.
for $t = 1, \ldots, c$ **do**
  Pick $i_t \in [n]$, where $\mathbf{Pr}(i_t = i) = p_i$;
  $S_{i_t t} = 1$;
  $D_{tt} = 1/\sqrt{cp_{i_t}}$.
end
$C = ASD$.

Algorithm 4. The EXACTLY(c) algorithm to create S, D, and C.

Data : $A \in \mathbb{R}^{m\times n}$, $p_i \geq 0, i \in [n]$ s.t. $\sum_{i\in[n]} p_i = 1$, positive integer $c \leq n$.
Result : Sampling matrix S, rescaling matrix D, and sampled and rescaled columns C.
Initialize S and D to the all zeros matrices.
$t = 1$;
for $j = 1, \ldots, n$ **do**
  Pick j with probability $\min\{1, cp_j\}$;
  if *j is picked* **then**
    $S_{jt} = 1$;
    $D_{tt} = 1/\min\{1, \sqrt{cp_j}\}$;
    $t = t + 1$;
  end
end
$C = ASD$.

Algorithm 5. *The* EXPECTED(c) *algorithm to create S, D, and C.*

A.2. Approximate matrix multiplication algorithms. Algorithm 6 takes as input two matrices A and B, a number $c \leq n$, and a probability distribution $\{p_i\}_{i=1}^n$ over $[n]$. It returns as output two matrices C and R, where the columns of C are a small number of sampled and rescaled columns of A and where the rows of R are a small number of sampled and rescaled rows of B. The sampling and rescaling are performed by calling either the EXACTLY(c) algorithm or the EXPECTED(c) algorithm. When the EXACTLY(c) algorithm is used to choose column-row pairs in Algorithm 6, this is identical to the algorithm of [21]. In particular, note that exactly c column-row pairs are chosen, and a column-row pair could be included in the sample more than once. When the EXPECTED(c) algorithm is used to choose column-row pairs in Algorithm 6, this is a minor variation of the algorithm of [21]. In particular, the

main difference is that at most c column-row pairs are chosen, and no column-row pair is included in the sample more than once.

Data : $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $\{p_i\}_{i=1}^n$ such that $\sum_{i=1}^n p_i = 1$, $c \leq n$.
Result : $C \in \mathbb{R}^{m \times c}$, $R \in \mathbb{R}^{c \times p}$.

- Form the matrix $C = ASD$ by sampling according to $\{p_i\}_{i=1}^n$ with the EXACTLY(c) algorithm or with the EXPECTED(c) algorithm;
- Form the matrix $R = DS^T B$ from the corresponding rows of B.

Algorithm 6. A fast Monte-Carlo algorithm for approximate matrix multiplication.

The next two theorems are our basic quality-of-approximation results for Algorithm 6. Each states that, under appropriate assumptions, $CR = ASDDS^T B \approx AB$. The most interesting of these assumptions is that the sampling probabilities used to randomly sample the columns of A and the corresponding rows of B are nonuniform and depend on the product of the Euclidean norms of the columns of A and/or the corresponding rows of B. For example, consider sampling probabilities $\{p_i\}_{i=1}^n$ such that

$$p_i \geq \beta \frac{\left|A^{(i)}\right|_2 \left|B_{(i)}\right|_2}{\sum_{j=1}^n \left|A^{(j)}\right|_2 \left|B_{(j)}\right|_2}, \tag{44}$$

for some $\beta \in (0, 1]$. Sampling probabilities of the form (44) use information from the matrices A and B in a very particular manner. If $\beta = 1$, they are optimal for approximating AB by CR in a sense made precise in [21]. Alternatively, sampling probabilities $\{p_i\}_{i=1}^n$ such that

$$p_i \geq \beta \frac{\left|A^{(i)}\right|_2^2}{\|A\|_F^2}, \tag{45}$$

for some $\beta \in (0, 1]$, are also of interest in approximating the product AB by CR if, e.g., only information about A is easily available.

The following theorem is our main quality-of-approximation result for approximating the product of two matrices with Algorithm 6, when column-row pairs are sampled using the EXACTLY(c) algorithm. Its proof (and the statement and proof of similar stronger results) may be found in [21].

THEOREM 6. Suppose $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $c \leq n$, and C, R are formed by Algorithm 6 using EXACTLY(c) and $\{p_i\}_{i=1}^n$ such that (44) or (45) holds. Then

$$\mathbf{E}\left[\|AB - CR\|_F\right] \leq \frac{1}{\sqrt{\beta c}} \|A\|_F \|B\|_F.$$

The following theorem is our main quality-of-approximation result for approximating the product of two matrices with Algorithm 6, when column-row pairs are sampled using the EXPECTED(c) algorithm. The Frobenius norm bound (46) is new, and the spectral norm bound (47) is due to Rudelson and Vershynin, who proved a similar result in a more general setting [54, 62, 56].

THEOREM 7. [illegible] $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$ [illegible] $c \leq n$ [illegible] C [illegible] R [illegible] 6 [illegible] EXPECTED(c) [illegible] $\{p_i\}_{i=1}^n$ [illegible] (44) [illegible] (45) [illegible]

$$\mathbf{E}\left[\,\|AB - CR\|_F\right] \leq \frac{1}{\sqrt{\beta c}}\,\|A\|_F\,\|B\|_F\,. \tag{46}$$

[illegible] $B = A^T$ [illegible]

$$\mathbf{E}\left[\,\left\|AA^T - CC^T\right\|_2\right] \leq O(1)\sqrt{\frac{\log c}{\beta c}}\,\|A\|_F\,\|A\|_2\,. \tag{47}$$

[illegible]. Equation (47) follows from the analysis of Rudelson and Vershynin, who considered spectral norm bounds on approximating the product of two matrices [54, 62, 56]. Note that they considered approximating the product AA^T by sampling with respect to probabilities of the form (45) with $\beta = 1$, but the analysis for general $\beta \in (0, 1]$ is analogous.

Next, we prove that for any set of probabilities $\{p_i\}_{i=1}^n$ the following holds:

$$\mathbf{E}\left[\,\|AB - CR\|_F^2\right] \leq \frac{1}{c}\sum_{j=1}^{n}\frac{\left|A^{(j)}\right|_2^2\left|B_{(j)}\right|_2^2}{p_j}. \tag{48}$$

Equation (46) follows from (48) by using Jensen's inequality and using the form of the sampling probabilities (44) and (45).

To establish (48), recall that the sampling is performed with the EXPECTED(c) algorithm. Let I_j, $j \in [n]$ be the indicator variable that is set to 1 if the jth column of A and the jth row of B are sampled (with probability $\min\{1, cp_j\}$) and is set to 0 otherwise. Recall that if $I_j = 1$, we scale both the jth column of A and the jth row of B by $1/\sqrt{\min\{1, cp_j\}}$. Thus,

$$\begin{aligned}\|AB - CR\|_F^2 &= \left\|AB - ASDDS^TB\right\|_F^2 \\ &= \left\|\sum_{j=1}^{n}\left(1 - \frac{I_j}{\min\{1, cp_j\}}\right)A^{(j)}B_{(j)}\right\|_F^2. \end{aligned}\tag{49}$$

Clearly, if $\min\{1, cp_j\} = 1$, then $I_j = 1$ with probability 1, and $1 - I_j/\min\{1, cp_j\} = 0$. Thus, we can focus on the set of indices $\Lambda = \{j \in [n] : cp_j < 1\} \subseteq [n]$. By taking the expectation of both sides of (49), it follows that

$$\begin{aligned}\mathbf{E}\left[\,\|AB - CR\|_F^2\right] &= \mathbf{E}\left[\left\|\sum_{j\in\Lambda}\left(1 - \frac{I_j}{cp_j}\right)A^{(j)}B_{(j)}\right\|_F^2\right] \\ &= \mathbf{E}\left[\sum_{i_1=1}^{m}\sum_{i_2=1}^{p}\left(\sum_{j\in\Lambda}\left(1 - \frac{I_j}{cp_j}\right)A^{(j)}B_{(j)}\right)_{i_1i_2}^2\right] \\ &= \mathbf{E}\left[\sum_{i_1=1}^{m}\sum_{i_2=1}^{p}\left(\sum_{j\in\Lambda}\left(1 - \frac{I_j}{cp_j}\right)A_{i_1j}B_{ji_2}\right)^2\right]. \end{aligned}$$

By multiplying out the right-hand side, it follows that

(50)
$$\mathbf{E}\left[\,\|AB-CR\|_F^2\right] = \mathbf{E}\left[\sum_{i_1=1}^{m}\sum_{i_2=1}^{p}\sum_{j_1\in\Lambda}\sum_{j_2\in\Lambda}\left(1-\frac{I_{j_1}}{cp_{j_1}}\right)\left(1-\frac{I_{j_2}}{cp_{j_2}}\right)A_{i_1j_1}B_{j_1i_2}A_{i_1j_2}B_{j_2i_2}\right]$$
$$= \sum_{i_1=1}^{m}\sum_{i_2=1}^{p}\sum_{j_1\in\Lambda}\sum_{j_2\in\Lambda}\mathbf{E}\left[\left(1-\frac{I_{j_1}}{cp_{j_1}}\right)\left(1-\frac{I_{j_2}}{cp_{j_2}}\right)\right]A_{i_1j_1}B_{j_1i_2}A_{i_1j_2}B_{j_2i_2}.$$

Notice that for $j\in[\Lambda]$, $\mathbf{E}\left[1-I_j/cp_j\right]=0$ and $\mathbf{E}\left[\left(1-I_j/cp_j\right)^2\right]=(1/cp_j)-1\le 1/cp_j$. Hence,

$$\mathbf{E}\left[\,\|AB-CR\|_F^2\right] = \sum_{i_1=1}^{m}\sum_{i_2=1}^{p}\sum_{j\in\Lambda}\mathbf{E}\left[\left(1-I_j/cp_j\right)^2\right]A_{i_1j}^2B_{ji_2}^2$$
$$\le \sum_{j\in\Lambda}\frac{1}{cp_j}\sum_{i_1=1}^{m}\sum_{i_2=1}^{p}A_{i_1j}^2B_{ji_2}^2 = \frac{1}{c}\sum_{j\in\Lambda}\frac{\left|A^{(j)}\right|_2^2\left|B_{(j)}\right|_2^2}{p_j}.$$

This concludes the proof of (48) and thus of the theorem. □

Acknowledgments. We would like to thank Sariel Har-Peled for writing up his results amidst travel in India [42], Amit Deshpande and Santosh Vempala for graciously providing a copy of [18], and two anonymous reviewers for useful comments.

REFERENCES

[1] D. Achlioptas and F. McSherry, *Fast computation of low rank matrix approximations*, J. ACM, 54 (2007).
[2] D. Achlioptas and F. McSherry, *Fast computation of low rank matrix approximations*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, 2001, pp. 611–6181.
[3] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan, *Approximating extent measures of points*, J. ACM, 51 (2004), pp. 606–635.
[4] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan, *Geometric approximation via coresets survey*, in Current Trends in Combinatorial and Computational Geometry, E. Welzl, ed., Cambridge University Press, Cambridge, 2006.
[5] N. Ailon and B. Chazelle, *Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform*, in Proceedings of the 38th Annual ACM Symposium on Theory of Computing, 2006, pp. 557–563.
[6] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, Springer-Verlag, New York, 2003.
[7] Y. Bengio, J. F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, *Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering*, in Annual Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference, 2004, pp. 177–184.
[8] M. W. Berry, S. A. Pulatova, and G. W. Stewart, *Computing sparse reduced-rank approximations to sparse matrices*, ACM Trans. Math. Software, 31 (2005), pp. 252–269.
[9] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.
[10] A. L. Blum and P. Langley, *Selection of relevant features and examples in machine learning*, Artif. Intell., 97 (1997), pp. 245–271.
[11] P. Businger and G. H. Golub, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–276.
[12] T. F. Chan and P. C. Hansen, *Computing truncated singular value decomposition least squares solutions by rank revealing QR-factorizations*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 519–530.

[13] T. F. Chan and P. C. Hansen, *Some applications of the rank revealing QR factorization*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 727–741.

[14] M. Charikar, V. Guruswami, R. Kumar, S. Rajagopalan, and A. Sahai, *Combinatorial feature selection problems*, in Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, 2000, pp. 631–642.

[15] The International HapMap Consortium, *A haplotype map of the human genome*, Nature, 437 (2005), pp. 1299–1320.

[16] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, and M. W. Mahoney, *Feature selection methods for text classification*, in Proceedings of the 13th Annual ACM SIGKDD Conference, 2007, pp. 230–239.

[17] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang, *Matrix approximation and projective clustering via volume sampling*, in Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, 2006, pp. 1117–1126.

[18] A. Deshpande and S. Vempala, *Adaptive sampling and fast low-rank matrix approximation*, Technical report TR06-042, Electronic Colloquium on Computational Complexity, 2006.

[19] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, *Clustering in large graphs and matrices*, in Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms, 1999, pp. 291–299.

[20] P. Drineas and R. Kannan, *Pass efficient algorithms for approximating large matrices*, in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 223–232.

[21] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices* I: *Approximating matrix multiplication*, SIAM J. Comput., 36 (2006), pp. 132–157.

[22] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices* II: *Computing a low-rank approximation to a matrix*, SIAM J. Comput., 36 (2006), pp. 158–183.

[23] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices* III: *Computing a compressed approximate matrix decomposition*, SIAM J. Comput., 36 (2006), pp. 184–206.

[24] P. Drineas and M. W. Mahoney, *Approximating a Gram matrix for improved kernel-based learning*, in Proceedings of the 18th Annual Conference on Learning Theory, 2005, pp. 323–337.

[25] P. Drineas and M. W. Mahoney, *On the Nyström method for approximating a Gram matrix for improved kernel-based learning*, J. Mach. Learn. Res., 6 (2005), pp. 2153–2175.

[26] P. Drineas and M. W. Mahoney, *A randomized algorithm for a tensor-based generalization of the Singular Value Decomposition*, Linear Algebra Appl., 420 (2007), pp. 553–571.

[27] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Polynomial time algorithm for column-row based relative-error low-rank matrix approximation*, manuscript, 2005.

[28] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Polynomial time algorithm for column-row based relative-error low-rank matrix approximation*, Technical report 2006-04, DIMACS, 2006.

[29] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Sampling algorithms for ℓ_2 regression and applications*, in Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, 2006, pp. 1127–1136.

[30] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Subspace sampling and relative-error matrix approximation: Column-based methods*, in Proceedings of the 10th International Workshop on Randomization and Computation, 2006, pp. 316–326.

[31] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Subspace sampling and relative-error matrix approximation: Column-row-based methods*, in Proceedings of the 14th Annual European Symposium on Algorithms, 2006, pp. 304–314.

[32] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, *Spectral grouping using the Nyström method*, IEEE Trans. Pattern Anal. Mach. Intell., 26 (2004), pp. 214–225.

[33] A. Frieze, R. Kannan, and S. Vempala, *Fast Monte-Carlo algorithms for finding low-rank approximations*, in Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, 1998, pp. 370–378.

[34] A. Frieze, R. Kannan, and S. Vempala, *Fast Monte-Carlo algorithms for finding low-rank approximations*, J. ACM, 51 (2004), pp. 1025–1041.

[35] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, *Eigentaste: A constant time collaborative filtering algorithm*, Inform. Retrieval, 4 (2001), pp. 133–151.

[36] G. Golub, *Numerical methods for solving linear least squares problems*, Numer. Math., 7 (1965), pp. 206–216.

[37] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996.
[38] S. A. Goreinov and E. E. Tyrtyshnikov, *The maximum-volume concept in approximation by low-rank matrices*, Contemp. Math., 280 (2001), pp. 47–51.
[39] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21.
[40] M. Gu and S. C. Eisenstat, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.
[41] I. Guyon and A. Elisseeff, *An introduction to variable and feature selection*, J. Mach. Learn. Res., 3 (2003), pp. 1157–1182.
[42] S. Har-Peled, *Low rank matrix approximation in linear time*, manuscript, 2006.
[43] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.
[44] F. G. Kuruvilla, P. J. Park, and S. L. Schreiber, *Vector algebra in the analysis of genome-wide expression data*, Genome Biol., 3 (2002), research0011.1–0011.11.
[45] S. Lafon, *Diffusion Maps and Geometric Harmonics*, Ph.D. thesis, Yale University, 2004.
[46] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, *RCV1: A new benchmark collection for text categorization research*, J. Mach. Learn. Res., 5 (2004), pp. 361–397.
[47] Z. Lin and R. B. Altman, *Finding haplotype tagging SNPs by use of principal components analysis*, Amer. J. Human Genetics, 75 (2004), pp. 850–861.
[48] M. W. Mahoney, M. Maggioni, and P. Drineas, *Tensor-CUR decompositions for tensor-based data*, in Proceedings of the 12th Annual ACM SIGKDD Conference, 2006, pp. 327–336.
[49] P.-G. Martinsson, V. Rokhlin, and M. Tygert, *A randomized algorithm for the approximation of matrices*, Technical report YALEU/DCS/TR-1361, Yale University Department of Computer Science, New Haven, CT, 2006.
[50] M. Z. Nashed, ed., *Generalized Inverses and Applications*, Academic Press, New York, 1976.
[51] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics in Applied Mathematics, SIAM, Philadelphia, 1998.
[52] P. Paschou, M. W. Mahoney, A. Javed, J. R. Kidd, A. J. Pakstis, S. Gu, K. K. Kidd, and P. Drineas, *Intra- and interpopulation genotype reconstruction from tagging SNPs*, Genome Res., 17 (2007), pp. 96–107.
[53] L. Rademacher, S. Vempala, and G. Wang, *Matrix approximation and projective clustering via iterative sampling*, Technical report MIT-LCS-TR-983, Massachusetts Institute of Technology, Cambridge, MA, 2005.
[54] M. Rudelson, *Random vectors in the isotropic position*, J. Funct. Anal., 164 (1999), pp. 60–72.
[55] M. Rudelson and R. Vershynin, *Approximation of matrices*, manuscript.
[56] M. Rudelson and R. Vershynin, *Sampling from large matrices: An approach through geometric functional analysis*, J. ACM, 54 (2007).
[57] T. Sarlós, *Improved approximation algorithms for large matrices via random projections*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, 2006, pp. 143–152.
[58] G. W. Stewart, *Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix*, Numer. Math., 83 (1999), pp. 313–323.
[59] G. W. Stewart, *Error analysis of the quasi-Gram-Schmidt algorithm*, Technical report UMIACS TR-2004-17 CMSC TR-4572, University of Maryland, College Park, MD, 2004.
[60] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
[61] E. Tyrtyshnikov, *Incomplete cross approximation in the mosaic-skeleton method*, Comput., 64 (2000), pp. 367–380.
[62] R. Vershynin, *Coordinate restrictions of linear operators in* l_2^n, manuscript.
[63] C. K. I. Williams, C. E. Rasmussen, A. Schwaighofer, and V. Tresp, *Observations on the Nyström method for Gaussian process prediction*, Technical report, University of Edinburgh, 2002.
[64] C. K. I. Williams and M. Seeger, *Using the Nyström method to speed up kernel machines*, in Annual Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference, 2002, pp. 682–688.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 882–897

CHARACTERIZATIONS OF A CLASS OF MATRICES AND PERTURBATION OF THE DRAZIN INVERSE*

N. CASTRO-GONZÁLEZ†, J. ROBLES†, AND J. Y. VÉLEZ-CERRADA†

Abstract. Given a singular square matrix A with index r, $\mathrm{ind}(A) = r$, we establish several characterizations in the Drazin inverse framework of the class of matrices B, which satisfy the conditions $\mathcal{N}(B^s)\cap\mathcal{R}(A^r) = \{0\}$ and $\mathcal{R}(B^s)\cap\mathcal{N}(A^r) = \{0\}$ with $\mathrm{ind}(B) = s$, where $\mathcal{N}(A)$ and $\mathcal{R}(A)$ denote the null space and the range space of a matrix A, respectively. We give explicit representations for B^{D} and BB^{D} and upper bounds for the errors $\|B^{\mathrm{D}} - A^{\mathrm{D}}\|/\|A^{\mathrm{D}}\|$ and $\|BB^{\mathrm{D}} - AA^{\mathrm{D}}\|$. In a numerical example we show that our bounds are better than others given in the literature.

Key words. singular matrix, Drazin inverse, eigenprojectors, perturbation

AMS subject classifications. 15A09, 15A23, 15A60, 65F35

DOI. 10.1137/060653366

1. Introduction and preliminaries.

Let $A \in \mathbb{C}^{n\times n}$ be any complex square matrix of order n with $\mathrm{ind}(A) = r$, where $\mathrm{ind}(A)$, the index of A, is the smallest nonnegative integer r such that $\operatorname{rank} A^r = \operatorname{rank} A^{r+1}$. Let $\mathcal{R}(A)$ and $\mathcal{N}(A)$ denote the range space of A and the null space of A, respectively. In our development we consider matrices $B \in \mathbb{C}^{n\times n}$, which satisfy the following condition for some positive integer s:

$$(\mathcal{C}_s) \qquad \mathcal{R}(B^s) \cap \mathcal{N}(A^r) = \{0\} \quad \text{and} \quad \mathcal{N}(B^s) \cap \mathcal{R}(A^r) = \{0\}.$$

A particular case is when the matrix B satisfies

$$\mathcal{R}(B^s) = \mathcal{R}(A^r) \quad \text{and} \quad \mathcal{N}(B^s) = \mathcal{N}(A^r). \tag{1.1}$$

The class of perturbation matrices B related to A by the condition (1.1), which is equivalent to the fact that both matrices have equal eigenprojection at zero, $B^\pi = A^\pi$ with $A^\pi = I - AA^{\mathrm{D}}$, were characterized in [4]. The Drazin inverse of B satisfying (1.1) is given by the formula $B^{\mathrm{D}} = (I + A^{\mathrm{D}}(B - A))^{-1}A^{\mathrm{D}}$. This latter formula was given in [15] for $B = A + E$, where $E = AA^{\mathrm{D}}EAA^{\mathrm{D}}$ and E sufficiently small.

The first and third authors gave in [5] characterizations of the matrices B related to A by the condition that, involving the eigenprojections at zero, $I - (B^\pi - A^\pi)^2$ is nonsingular. Therein, it was proved that $B^{\mathrm{D}} = (I + A^{\mathrm{D}}(B - A) + S)^{-1}A^{\mathrm{D}}(I - S)$ where $S = B^\pi - A^\pi$ and an upper bound for $\|B^{\mathrm{D}} - A^{\mathrm{D}}\|/\|A^{\mathrm{D}}\|$ was given in terms of $\|A^{\mathrm{D}}(B - A)\|$ and $\|B^\pi - A^\pi\|$.

The continuity of the Drazin inverse was studied in [1, 2, 3, 11]. In [2], Campbell and Meyer established that if A_j converges to A, then A_j^{D} converges to A^{D} if and only if $\operatorname{rank} A_j^{r_j} = \operatorname{rank} A^r$ for all sufficiently large j, where $r_j = \mathrm{ind}(A_j)$. Recently, the perturbation of the Drazin inverse was studied by several authors, and upper bounds for the relative error $\|B^{\mathrm{D}} - A^{\mathrm{D}}\|/\|A^{\mathrm{D}}\|$ were given under certain conditions [4, 5, 6, 8, 9, 12, 13, 14, 15, 16].

*Received by the editors March 2, 2006; accepted for publication (in revised form) by Z. Bai May 5, 2008; published electronically September 17, 2008. This research was partly supported by project MTM2007-67232, "Ministerio de Educación y Ciencia" of Spain.
http://www.siam.org/journals/simax/30-2/65336.html

†Facultad de Informática, Universidad Politécnica de Madrid, 28660 Boadilla del Monte, Madrid, Spain (nieves@fi.upm.es, jrobles@fi.upm.es, jyvelezc@hotmail.com).

In this paper, in section 2 we prove that, for a matrix B with $\mathrm{ind}(B) = s$, the fact that B satisfies condition $(\mathcal{C}_s)$ is equivalent to that $I-(B^\pi - A^\pi)^2$ is nonsingular. We establish several new characterizations of the matrices which satisfy condition $(\mathcal{C}_s)$. In terms of matrix rank, this class of matrices is characterized by the condition $\operatorname{rank} A^r = \operatorname{rank} B^s = \operatorname{rank} A^r B^s A^r$ whenever $s = \mathrm{ind}(B)$.

In section 3 we study further characterizations for the class $(\mathcal{C}_1)$, giving a representation of matrices $B \in (\mathcal{C}_1)$ such that $\mathrm{ind}(B) = 1$, with respect to the core-nilpotent block form of the matrix A. We mention that the perturbation of the group inverse is a case of special interest due to its application to stability of Markov chains [3, 10].

In section 4 we extend the characterizations for the group inverse to the general case of perturbations satisfying condition $(\mathcal{C}_s)$. We give an expression for the index 1-nilpotent decomposition of the matrices $B \in (\mathcal{C}_s)$, $\mathrm{ind}(B) = s$, which will be the main tool in the development of perturbation results.

Finally, in section 5 we give an explicit representation of B^{D}, and we derive upper bounds for the errors $\|B^{\mathrm{D}} - A^{\mathrm{D}}\|/\|A^{\mathrm{D}}\|$ and $\|BB^{\mathrm{D}} - AA^{\mathrm{D}}\|$ in terms of norms involving the powers $B^s - A^s$. In a numerical example we compare our bounds with others given recently in [13, 14].

In relation to the study of the continuity of the Drazin inverse, we can say that if A_j converges to A and $\operatorname{rank} A_j^{r_j} = \operatorname{rank} A^r A_j^{r_j} A^r = \operatorname{rank} A^r$ for all sufficiently large j, where $r_j = \mathrm{ind}(A_j)$, then an explicit representation for A_j^{D} and an explicit error bound of $\|A_j^{\mathrm{D}} - A^{\mathrm{D}}\|/\|A^{\mathrm{D}}\|$ are provided.

We recall that the [illegible] $A \in \mathbb{C}^{n\times n}$ is the unique matrix $A^{\mathrm{D}} \in \mathbb{C}^{n\times n}$ satisfying the relations

$$A^{\mathrm{D}} A A^{\mathrm{D}} = A^{\mathrm{D}}, \quad AA^{\mathrm{D}} = A^{\mathrm{D}} A, \quad A^{l+1} A^{\mathrm{D}} = A^l \quad \text{for all } l \geq r,$$

where $r = \mathrm{ind}(A)$. If A is nonsingular, then $\mathrm{ind}(A) = 0$ and the solution to the above equations is $A^{\mathrm{D}} = A^{-1}$. The case when $\mathrm{ind}(A) = 1$, i.e., $\operatorname{rank} A = \operatorname{rank} A^2$, the Drazin inverse is called the [illegible] A and is denoted by $A^\sharp$.

We denote by O a null matrix. Each $A \in \mathbb{C}^{n\times n}$ with $\mathrm{ind}(A) = r$ has a unique [illegible] (see [1, Theorem 11, Chapter 4]),

$$A = C_A + N_A, \quad \mathrm{ind}(C_A) = 1, \quad C_A N_A = N_A C_A = O, \quad N_A^r = O. \tag{1.2}$$

Moreover, we have $A^k = C_A^k + N_A^k$ for all integers $k \geq 1$, and $A^{\mathrm{D}} = C_A^\sharp$.

The following lemma gives a condition for the existence of the group inverse of a partitioned matrix and a formula for its computation (see [3, Theorems 7.7.5 and 7.7.7]).

LEMMA 1.1. [illegible] $M = \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)$ [illegible] $A \in \mathbb{C}^{d\times d}$ [illegible] $\Psi = I + A^{-1}BCA^{-1}$. [illegible]

(i) $\operatorname{rank} M = \operatorname{rank} A \Longleftrightarrow D = CA^{-1}B$.

[illegible] $k \geq 1$, M^k [illegible]

$$M^k = \begin{bmatrix} I \\ CA^{-1} \end{bmatrix} (A\Psi)^{k-1} A \begin{bmatrix} I & A^{-1}B \end{bmatrix}. \tag{1.3}$$

(ii) [illegible] $\operatorname{rank} M = \operatorname{rank} A$ [illegible] $\mathrm{ind}(M) = 1 \Longleftrightarrow \Psi$ [illegible] M [illegible]

$$M^\sharp = \begin{bmatrix} I \\ CA^{-1} \end{bmatrix} (\Psi A \Psi)^{-1} \begin{bmatrix} I & A^{-1}B \end{bmatrix}. \tag{1.4}$$

Let $A \in \mathbb{C}^{n\times n}$ with $\operatorname{ind}(A) = r$. The [illegible] A [illegible] 0, denoted by A^π, is the uniquely determined [illegible] such that $\mathcal{R}(A^\pi) = \mathcal{N}(A^r)$ and $\mathcal{N}(A^\pi) = \mathcal{R}(A^r)$.

If $\operatorname{ind}(A) = r > 0$, then there exists a nonsingular matrix P such that we can write A in the [illegible]

$$A = P \begin{pmatrix} A_1 & O \\ O & A_2 \end{pmatrix} P^{-1} \quad A_1 \in \mathbb{C}^{d\times d} \text{ nonsingular},\ d = \operatorname{rank} A^r,\ A_2^r = O. \tag{1.5}$$

By [3, Theorem 7.2.1], relative to the form (1.5), the Drazin inverse of A and the eigenprojection of A at zero are given by

$$A^{\mathrm{D}} = P \begin{pmatrix} A_1^{-1} & O \\ O & O \end{pmatrix} P^{-1}, \quad A^\pi = I - AA^{\mathrm{D}} = P \begin{pmatrix} O & O \\ O & I \end{pmatrix} P^{-1}.$$

The case when $\operatorname{ind}(A) = 1$ is equivalent to having $A_2 = O$ in (1.5), and so $A^\pi A = AA^\pi = O$. Moreover, we have $\mathcal{N}(A^\pi) = \mathcal{R}(A)$ and $\mathcal{R}(A^\pi) = \mathcal{N}(A)$.

LEMMA 1.2. [illegible] $A, C \in \mathbb{C}^{n\times n}$ [illegible] $\operatorname{ind}(A) = r$ [illegible] C [illegible]

$$I - A^\pi + CA^\pi C^{-1} A^\pi \text{ [illegible]} \iff I - A^\pi + C^{-1} A^\pi C A^\pi \text{ [illegible]}.$$

[illegible]. Write

$$C = P \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} P^{-1} \quad \text{and} \quad C^{-1} = P \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} P^{-1},$$

where C_{11}, X_{11}, and A_1 as in (1.5) are the same size. Then

$$I - A^\pi + CA^\pi C^{-1} A^\pi = P \begin{pmatrix} I & C_{12}X_{22} \\ O & C_{22}X_{22} \end{pmatrix} P^{-1},$$

$$I - A^\pi + C^{-1} A^\pi C A^\pi = P \begin{pmatrix} I & X_{12}C_{22} \\ O & X_{22}C_{22} \end{pmatrix} P^{-1}.$$

Hence, since $C_{22}X_{22}$ is nonsingular $\iff$ $X_{22}C_{22}$ is nonsingular, the equivalence given in this lemma follows. ∎

The following lemma is concerned with the rank of a product of matrices (see [17, sec. 2.4]).

LEMMA 1.3. [illegible] $A, B, C \in \mathbb{C}^{n\times n}$ [illegible]

$$\operatorname{rank} AB = \operatorname{rank} B - \dim(\mathcal{R}(B) \cap \mathcal{N}(A)), \tag{1.6}$$

$$\operatorname{rank} ABC \geq \operatorname{rank} AB + \operatorname{rank} BC - \operatorname{rank} B. \tag{1.7}$$

2. Characterizations of matrices satisfying condition ($\mathcal{C}_s$). First, for a matrix B with $\mathrm{ind}(B) = s$ we establish the equivalence among condition ($\mathcal{C}_s$) and conditions involving the matrix rank, and other conditions expressed in terms of the eigenprojections at zero.

THEOREM 2.1. [illegible] $A \in \mathbb{C}^{n\times n}$, $\mathrm{ind}(A) = r$ [illegible] $B \in \mathbb{C}^{n\times n}$ [illegible] $\mathrm{ind}(B) = s$ [illegible]

(a) B [illegible] ($\mathcal{C}_s$)

(b) $\mathrm{rank}\, B^s = \mathrm{rank}\, A^r = \mathrm{rank}\, A^r B^s = \mathrm{rank}\, B^s A^r$

(c) $\mathrm{rank}\, B^s = \mathrm{rank}\, A^r = \mathrm{rank}\, A^r B^s A^r$

(d) $\mathrm{rank}\, B^s = \mathrm{rank}\, A^r$, $I - A^\pi + B^\pi A^\pi$ [illegible]

(e) $I - (B^\pi - A^\pi)^2$ [illegible]

(f) $I - B^\pi - A^\pi$ [illegible]

[illegible] (a) $\Rightarrow$ (b). From the space decomposition $\mathbb{C}^n = \mathcal{R}(A^r) \oplus \mathcal{N}(A^r) = \mathcal{R}(B^s) \oplus \mathcal{N}(B^s)$ and the conditions $\mathcal{N}(B^s) \cap \mathcal{R}(A^r) = \{0\}$ and $\mathcal{R}(B^s) \cap \mathcal{N}(A^r) = \{0\}$, it is clear that $\mathrm{rank}\, B^s = \mathrm{rank}\, A^r$. Moreover, using Lemma 1.3, identity (1.6), we get

$$\mathrm{rank}\, A^r B^s = \mathrm{rank}\, B^s - \dim \mathcal{R}(B^s) \cap \mathcal{N}(A^r)$$

and

$$\mathrm{rank}\, B^s A^r = \mathrm{rank}\, A^r - \dim \mathcal{R}(A^r) \cap \mathcal{N}(B^s).$$

Hence, $\mathrm{rank}\, A^r B^s = \mathrm{rank}\, B^s$ and $\mathrm{rank}\, B^s A^r = \mathrm{rank}\, A^r$. So, (b) is proved.

(b) $\Rightarrow$ (c). Applying Lemma 1.3, formula (1.7), we get

$$\mathrm{rank}\, A^r B^s A^r \geq \mathrm{rank}\, A^r B^s + \mathrm{rank}\, B^s A^r - \mathrm{rank}\, B^s.$$

Hence $\mathrm{rank}\, A^r B^s A^r \geq \mathrm{rank}\, B^s$. We also have $\mathrm{rank}\, A^r B^s A^r \leq \mathrm{rank}\, A^r = \mathrm{rank}\, B^s$, so we conclude that $\mathrm{rank}\, A^r B^s A^r = \mathrm{rank}\, B^s$.

(c) $\Rightarrow$ (d). From condition $\mathrm{rank}\, A^r B^s A^r = \mathrm{rank}\, A^r = \mathrm{rank}\, B^s$, using Lemma 1.3, identity (1.6), we easily derive $\mathcal{R}(A^r) \cap \mathcal{N}(B^s) = \{0\}$ and $\mathcal{N}(A^r) \cap \mathcal{R}(B^s) = \{0\}$. Now, let $(I - A^\pi + B^\pi A^\pi)x = 0$. Then $(I - A^\pi)x = -B^\pi A^\pi x$. From this latter relation it follows that $(I - A^\pi)x \in \mathcal{R}(A^r) \cap \mathcal{N}(B^s)$, and thus $(I - A^\pi)x = 0$. Further, we also have $B^\pi A^\pi x = 0$. Hence $A^\pi x \in \mathcal{R}(B^s) \cap \mathcal{N}(A^r)$ and, consequently, $A^\pi x = 0$. Therefore $x = 0$, and $I - A^\pi + B^\pi A^\pi$ is nonsingular.

(d) $\Rightarrow$ (e). Since $I - (B^\pi - A^\pi)^2 = (I - A^\pi + B^\pi A^\pi)(I - B^\pi + A^\pi B^\pi)$, we have to prove that $I - B^\pi + A^\pi B^\pi$ is nonsingular. We write the core-nilpotent block forms, as in (1.5), $A = P\left(\begin{smallmatrix} A_1 & O \\ O & A_2 \end{smallmatrix}\right)P^{-1}$ and $B = Q\left(\begin{smallmatrix} B_1 & O \\ O & B_2 \end{smallmatrix}\right)Q^{-1}$ with A_1 and B_1 nonsingular matrices. We note that A_1 and B_1 have the same size because $\mathrm{rank}\, B^s = \mathrm{rank}\, A^r$. Moreover, $\left(Q^{-1}B^\pi Q = \left(\begin{smallmatrix} O & O \\ O & I \end{smallmatrix}\right)\right) = P^{-1}A^\pi P$ and, thus, $B^\pi = QP^{-1}A^\pi PQ^{-1}$. Hence $I - A^\pi + B^\pi A^\pi = I - A^\pi + QP^{-1}A^\pi PQ^{-1}A^\pi$. So $I - A^\pi + QP^{-1}A^\pi PQ^{-1}A^\pi$ is nonsingular, and by Lemma 1.2 we conclude that $PQ^{-1}(I - B^\pi + A^\pi B^\pi)QP^{-1} = I - A^\pi + PQ^{-1}A^\pi QP^{-1}A^\pi$ is also nonsingular.

(e) $\Rightarrow$ (f). Let $(I - B^\pi - A^\pi)x = 0$. Then $(I - B^\pi + A^\pi)x = 2A^\pi x$, and hence $(I + B^\pi - A^\pi)(I - B^\pi + A^\pi)x = 2B^\pi A^\pi x = 0$. So, we have $(I - (B^\pi - A^\pi)^2)x = 0$. This implies that $x = 0$, and therefore $I - B^\pi - A^\pi$ is nonsingular.

(f) $\Rightarrow$ (a). This equivalence follows from [7, Theorem 1.2], applying the equivalence of (iii) and (iv) given therein with the projectors $I - A^\pi$ and B^π. □

The next lemma gives properties that are needed in what follows.

LEMMA 2.2. [illegible] $A \in \mathbb{C}^{n\times n}$, $\mathrm{ind}(A) = r$ [illegible] $B \in \mathbb{C}^{n\times n}$ [illegible] $\mathrm{ind}(B) = s$ [illegible] ($\mathcal{C}_s$) [illegible]

(i) $l \geq s$, $I + (A^{\mathrm{D}})^l(B^l - A^l)$

(ii) $I - (I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi - A^\pi(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}$

. (i) Let $l \geq s$ and $(I+(A^{\mathrm{D}})^l(B^l-A^l))x = 0$. Then, $A^\pi x = -(A^{\mathrm{D}})^l B^l x = 0$. Hence, $x \in \mathcal{N}(A^\pi) = \mathcal{R}(A^r)$ and $B^l x \in \mathcal{N}\left((A^{\mathrm{D}})^l\right) = \mathcal{N}(A^r)$. Since $\mathcal{R}(B^l) = \mathcal{R}(B^s)$, then $B^l x \in \mathcal{R}(B^s) \cap \mathcal{N}(A^r)$. So, $B^l x = 0$. Therefore, $x \in \mathcal{N}\left(B^l\right) \cap \mathcal{R}(A^r)$, and thus $x = 0$. So, $I + (A^{\mathrm{D}})^l(B^l - A^l)$ is nonsingular.

(ii) Let $x - (I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi x - A^\pi(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}x = 0$. Then $(I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}(A^{\mathrm{D}})^s B^s x = A^\pi(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}x$. From this identity and the fact that $(I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}(A^{\mathrm{D}})^s = (A^{\mathrm{D}})^s(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}$, we conclude that $(I+(A^{\mathrm{D}})^s(B^s-A^s))^{-1}(A^{\mathrm{D}})^s B^s x = 0$ and $A^\pi(I+(B^s-A^s)(A^{\mathrm{D}})^s)^{-1}x = 0$. Therefore, $(A^{\mathrm{D}})^s B^s x = 0$ and so $B^s x \in \mathcal{R}(B^s) \cap \mathcal{N}(A^r)$. Thus, $B^s x = 0$. Moreover, since $(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}x \in \mathcal{R}(A^r)$, $(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}x = A^r y$ for some y. This implies that $x = B^s(A^{\mathrm{D}})^s A^r y$, and so $x \in \mathcal{R}(B^s) \cap \mathcal{N}(B^s)$. Hence, $x = 0$ because $\mathrm{ind}(B) = s$. So, (ii) is proved. ∎

In the following theorem, we will derive a formula for the eigenprojection of B at zero, B^π.

THEOREM 2.3. $A \in \mathbb{C}^{n\times n}$, $\mathrm{ind}(A) = r$, $B \in \mathbb{C}^{n\times n}$ ... $\mathrm{ind}(B) = s$ ... $(\mathcal{C}_s)$ ...

$$B^\pi = -(I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi X^{-1} = -X^{-1}A^\pi(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1},$$

...

$$X = I - (I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi - A^\pi(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}.$$

. From Lemma 2.2 we know that $I+(A^{\mathrm{D}})^s(B^s-A^s)$ and X are nonsingular. Using that $A^\pi(I+(A^{\mathrm{D}})^s(B^s-A^s))^{-1} = A^\pi = (I+(B^s-A^s)(A^{\mathrm{D}})^s)^{-1}A^\pi$, it is easily checked that

$$\begin{aligned} &X(I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi \\ &\quad = -A^\pi(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}(I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi \\ &\quad = A^\pi(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}X. \end{aligned} \tag{2.1}$$

Hence

$$(I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi X^{-1} = X^{-1}A^\pi(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}. \tag{2.2}$$

Let $Q = -(I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi X^{-1}$. We observe that

$$\mathcal{R}(Q) = \mathcal{R}((I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi)$$

because X is nonsingular. Let us show that Q is the projector with $\mathcal{N}(Q) = \mathcal{R}(B^s)$ and $\mathcal{R}(Q) = \mathcal{N}(B^s)$. First, using (2.2) and (2.1) we see that

$$Q^2 = X^{-1}A^\pi(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}(I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi X^{-1} = Q.$$

Now, let us assume that $x \in \mathcal{N}(B^s)$. Then $A^\pi x + (A^{\mathrm{D}})^s B^s x = A^\pi x$. From this relation it follows that $x = (A^\pi + (A^{\mathrm{D}})^s B^s)^{-1}A^\pi x$ and, thus, $x \in \mathcal{R}(Q)$. Conversely, assuming $x \in \mathcal{R}(Q)$ we get $(A^\pi + (A^{\mathrm{D}})^s B^s)x = A^\pi y$ for some $y \in \mathbb{C}^n$. Hence $(A^{\mathrm{D}})^s B^s x = A^\pi(y - x)$. Then $(A^{\mathrm{D}})^s B^s x = 0$. Therefore, $B^s x \in \mathcal{R}(B^s) \cap \mathcal{N}(A^r)$. So $B^s x = 0$. Consequently, $\mathcal{R}(Q) = \mathcal{N}(B^s)$.

By (2.2) we have that $\mathcal{N}(Q) = \mathcal{N}\left(X^{-1}A^{\pi}(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}\right)$. Hence it follows that $\mathcal{N}(Q) = \mathcal{N}\left(A^{\pi}(I + (B^s - A^s)(A^{\mathrm{D}})^s)^{-1}\right)$ because X is nonsingular. Let us assume that $x \in \mathcal{N}(Q)$. Then

$$A^{\pi}(A^{\pi} + B^s(A^{\mathrm{D}})^s)^{-1}x = (I - B^s(A^{\mathrm{D}})^s(A^{\pi} + B^s(A^{\mathrm{D}})^s)^{-1})x = 0.$$

Hence, $x = B^s A^{\mathrm{D}}(A^{\pi} + B^s(A^{\mathrm{D}})^s)^{-1}x$, and thus $x \in \mathcal{R}(B^s)$. Since $\mathcal{N}(Q) \subseteq \mathcal{R}(B^s)$, and $\mathbb{C}^n = \mathcal{R}(Q) \oplus \mathcal{N}(Q) = \mathcal{R}(B^s) \oplus \mathcal{N}(B^s)$ because $\mathrm{ind}(B) = s$, we conclude that $\mathcal{N}(Q) = \mathcal{R}(B^s)$. So we have $B^{\pi} = Q$, which is the desired result. □

3. The class $(\mathcal{C}_1)$. We shall first give further characterizations of matrices B satisfying condition $(\mathcal{C}_1)$ and $\mathrm{ind}(B) = 1$. We obtain a representation of B with respect to the core-nilpotent block form of the matrix A.

THEOREM 3.1. [illegible] $A \in \mathbb{C}^{n\times n}$, $\mathrm{ind}(A) = r$ [illegible] $B \in \mathbb{C}^{n\times n}$ [illegible]

(a) B [illegible] $(\mathcal{C}_1)$ [illegible] $\mathrm{ind}(B) = 1$

(b) $B(I + A^{\mathrm{D}}(B - A))^{-1}A^{\pi} = O$, $I + A^{\mathrm{D}}(B - A)$ [illegible] $I + (A^{\mathrm{D}})^2(B^2 - A^2)$ [illegible]

(c) [illegible] A [illegible] (1.5), B [illegible]

$$B = P\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{21}B_{11}^{-1}B_{12} \end{pmatrix}P^{-1}, \tag{3.1}$$

[illegible] B_{11} [illegible] $I + B_{11}^{-1}B_{12}B_{21}B_{11}^{-1}$ [illegible]

(d) $\mathrm{rank}\, B = \mathrm{rank}\, A^r$, $I + A^{\mathrm{D}}(B - A)$ [illegible] $I + (A^{\mathrm{D}})^2(B^2 - A^2)$ [illegible]

[illegible]. (a) ⇒ (b). Since $\mathrm{ind}(B) = 1$, from Lemma 2.2(i) we get that $I + A^{\mathrm{D}}(B - A)$ and $I + (A^{\mathrm{D}})^2(B^2 - A^2)$ are nonsingular. Finally, using that $BB^{\pi} = O$ and applying Theorem 2.3, we conclude that $B(I + A^{\mathrm{D}}(B - A))^{-1}A^{\pi} = O$.

(b) ⇒ (c). Write

$$B = P\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}P^{-1}.$$

We compute

$$I + A^{\mathrm{D}}(B - A) = P\begin{pmatrix} A_1^{-1}B_{11} & A_1^{-1}B_{12} \\ O & I \end{pmatrix}P^{-1}.$$

Hence B_{11} is nonsingular because $I + A^{\mathrm{D}}(B - A)$ is nonsingular. We have

$$I + (A^{\mathrm{D}})^2(B^2 - A^2) = P\begin{pmatrix} A_1^{-2}(B_{11}^2 + B_{12}B_{21}) & A_1^{-2}(B_{11}B_{12} + B_{12}B_{22}) \\ O & I \end{pmatrix}P^{-1}.$$

Thus, $B_{11}^2 + B_{12}B_{21}$ is nonsingular because $I + (A^{\mathrm{D}})^2(B^2 - A^2)$ is nonsingular. On the other hand,

$$\begin{aligned} B(I + A^{\mathrm{D}}(B - A))^{-1}A^{\pi} &= P\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}\begin{pmatrix} B_{11}^{-1}A_1 & -B_{11}^{-1}B_{12} \\ O & I \end{pmatrix}\begin{pmatrix} O & O \\ O & I \end{pmatrix}P^{-1} \\ &= P\begin{pmatrix} O & O \\ O & -B_{21}B_{11}^{-1}B_{12} + B_{22} \end{pmatrix}P^{-1}. \end{aligned}$$

From the assumption $B(I + A^{\mathrm{D}}(B - A))^{-1}A^{\pi} = O$ it follows that $B_{22} = B_{21}B_{11}^{-1}B_{12}$.

(c) $\Leftrightarrow$ (d). From the representation (3.1), applying Lemma 1.1, it follows that $\operatorname{rank} B = \operatorname{rank} B_{11} = \operatorname{rank} A^r$. The rest is easily seen.

Conversely, write

$$B = P \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} P^{-1}.$$

Since $I + A^{\mathrm{D}}(B - A)$ and $I + (A^{\mathrm{D}})^2(B^2 - A^2)$ are nonsingular, arguing as in the proof of (b) $\Rightarrow$ (c), we get that B_{11} and $I + B_{11}^{-1}B_{12}B_{21}B_{11}^{-1}$ are nonsingular. Finally, from $\operatorname{rank} B = \operatorname{rank} A^r$ we obtain that $\operatorname{rank} B = \operatorname{rank} B_{11}$, and hence by Lemma 1.1(i) we conclude that $B_{22} = B_{21}B_{11}^{-1}B_{12}$.

(c) $\Rightarrow$ (a). Assume that B has the block representation (3.1). By Lemma 1.1(i), (ii), we conclude that $\operatorname{rank} B = \operatorname{rank} B_{11} = \operatorname{rank} A^r$ and $\operatorname{ind}(B) = 1$. On the other hand,

$$\operatorname{rank} A^r B A^r = \operatorname{rank} P \begin{pmatrix} A_1^r B_{11} A_1^r & O \\ O & O \end{pmatrix} P^{-1} = \operatorname{rank} A_1^r B_{11} A_1^r = \operatorname{rank} A^r.$$

Hence, in view of Theorem 2.1 (a)$\Leftrightarrow$(c), we conclude that B satisfy condition $(\mathcal{C}_1)$. ∎

3.2. Conditions (b) and (d) in the above theorem can be replaced by the following symmetrical conditions:

(b′) $A^{\pi}(I + (B - A)A^{\mathrm{D}})^{-1}B = O$, $I + (B - A)A^{\mathrm{D}}$ and $I + (B^2 - A^2)(A^{\mathrm{D}})^2$ are nonsingular.

(d′) $\operatorname{rank} B = \operatorname{rank} A^r$, $I + (B - A)A^{\mathrm{D}}$ and $I + (B^2 - A^2)(A^{\mathrm{D}})^2$ are nonsingular.

Next, we state the following compact representation for B and $B^{\sharp}$.

LEMMA 3.3. [illegible] $A \in \mathbb{C}^{n\times n}$, $\operatorname{ind}(A) = r$ [illegible] $B \in \mathbb{C}^{n\times n}$, $\operatorname{ind}(B) = 1$ [illegible] $(\mathcal{C}_1)$ [illegible]

$$B = P \begin{bmatrix} I \\ S \end{bmatrix} B_1 \begin{bmatrix} I & T \end{bmatrix} P^{-1}, \tag{3.2}$$

[illegible] B_1 [illegible] $I+TS$ [illegible] B [illegible]

$$B^{\sharp} = P \begin{bmatrix} I \\ S \end{bmatrix} [(I + TS)B_1(I + TS)]^{-1} \begin{bmatrix} I & T \end{bmatrix} P^{-1}. \tag{3.3}$$

[illegible]. By Theorem 3.1 (c),

$$B = P \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{21}B_{11}^{-1}B_{12} \end{pmatrix} P^{-1},$$

where B_{11} and $I + B_{11}^{-1}B_{12}B_{21}B_{11}^{-1}$ are nonsingular. By denoting $B_1 = B_{11}$, $T = B_{11}^{-1}B_{12}$, and $S = B_{21}B_{11}^{-1}$ we get the representation (3.2). Now, applying formula (1.4) given in Lemma 1.1, we obtain the representation for $B^{\sharp}$. ∎

4. The class $(\mathcal{C}_s)$. Next, based on Theorem 3.1, we establish the following new characterizations of B satisfying condition $(\mathcal{C}_s)$.

THEOREM 4.1. [illegible] $A \in \mathbb{C}^{n\times n}$, $\operatorname{ind}(A) = r$ [illegible] $B \in \mathbb{C}^{n\times n}$ [illegible]

(a) B [illegible] $(\mathcal{C}_s)$ [illegible] $\operatorname{ind}(B) = s$

(b) [illegible] s [illegible] $B^s(I+(A^{\mathrm{D}})^s(B^s-A^s))^{-1}A^{\pi}=O$, $I+(A^{\mathrm{D}})^s(B^s-A^s)$ [illegible] $I+(A^{\mathrm{D}})^{s+1}(B^{s+1}-A^{s+1})$ [illegible]

(c) [illegible] 1 [illegible] B [illegible] A, (1.5),

$$B = C_B + N_B = P\begin{bmatrix} I \\ S \end{bmatrix} B_1 \begin{bmatrix} I & T \end{bmatrix} P^{-1} + P\begin{bmatrix} T \\ -I \end{bmatrix} B_2 \begin{bmatrix} S & -I \end{bmatrix} P^{-1}, \tag{4.1}$$

[illegible] B_1 [illegible] $I+TS$ [illegible] $B_2(I+ST)$ [illegible] s

(d) [illegible] s [illegible] $\operatorname{rank} B^s = \operatorname{rank} A^r$, $I+(A^{\mathrm{D}})^s(B^s-A^s)$ [illegible] $I+(A^{\mathrm{D}})^{s+1}(B^{s+1}-A^{s+1})$ [illegible]

. If $\operatorname{ind}(B)=s$ and B satisfies condition $(\mathcal{C}_s)$, then s is the smallest positive integer such that B^s satisfies condition $(\mathcal{C}_1)$ and $\operatorname{ind}(B^s)=1$. Moreover, we observe that for any $k\geq s$, $I+(A^{\mathrm{D}})^k(B^k-A^k)$ is nonsingular if and only if $I+A^{\mathrm{D}}(B^k-A)$ is nonsingular. So, applying Theorem 3.1 with B^s, it follows the equivalence between condition (a) and the following:

(b′) For the smallest positive integer s such that $B^s(I+(A^{\mathrm{D}})^s(B^s-A^s))^{-1}A^{\pi}=O$, $I+(A^{\mathrm{D}})^s(B^s-A^s)$ and $I+(A^{\mathrm{D}})^{2s}(B^{2s}-A^{2s})$ are nonsingular.

We now note that conditions (b′) and (b) are equivalent.

A similar device proves the equivalence between conditions (a) and (d) in this theorem. Applying Theorem 3.1 with B^s we get the equivalence of (a) and the following:

(d′) For the smallest positive integer s such that $\operatorname{rank} B^s = \operatorname{rank} A^r$, we have that $I+(A^{\mathrm{D}})^s(B^s-A^s)$ and $I+(A^{\mathrm{D}})^{2s}(B^{2s}-A^{2s})$ are nonsingular.

Finally, we note that conditions (d′) and (d) are equivalent.

Now, we will prove the equivalence between (a) and (c). Suppose $B=C_B+N_B$ is the index 1-nilpotent decomposition (1.2) of B. We know that if s is the index of B, then $\mathcal{N}(C_B)=\mathcal{N}(B^s)$ and $\mathcal{R}(C_B)=\mathcal{R}(B^s)$. Hence if B satisfies condition $(\mathcal{C}_s)$, then C_B satisfies condition $(\mathcal{C}_1)$ and $\operatorname{ind}(C_B)=1$. By Lemma 3.3 it follows that

$$C_B = P\begin{bmatrix} I \\ S \end{bmatrix} B_1 \begin{bmatrix} I & T \end{bmatrix} P^{-1}, \tag{4.2}$$

where B_1 and $I+TS$ are nonsingular. We observe that $I+ST$ is also nonsingular. Now, write

$$N_B = P\begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix} P^{-1}.$$

Since $C_BN_B=N_BC_B=O$, by direct computations it follows that $N_{11}=TN_{22}S$, $N_{12}=-TN_{22}$, and $N_{21}=-N_{22}S$. So,

$$N_B = P\begin{bmatrix} T \\ -I \end{bmatrix} B_2 \begin{bmatrix} S & -I \end{bmatrix} P^{-1}, \tag{4.3}$$

where we have renamed $B_2=N_{22}$. Thus, for every positive integer k,

$$N_B^k = P\begin{bmatrix} T \\ -I \end{bmatrix} (B_2(I+ST))^{k-1} B_2 \begin{bmatrix} S & -I \end{bmatrix} P^{-1}. \tag{4.4}$$

Condition $N_B^s=O$ implies that $(B_2(I+ST))^s=O$. Therefore, $B_2(I+ST)$ is nilpotent of index s. Hence, from (4.2) and (4.3) we get the representation (4.1).

Conversely, assume that we have the splitting $B = C_B + N_B$, where C_B and N_B have the representation given by (4.1). Clearly $C_B N_B = N_B C_B = O$. Moreover, by Theorem 3.1, equivalence between (a) and (c), it follows that C_B satisfies condition $(\mathcal{C}_1)$ and $\mathrm{ind}(C_B) = 1$. Using (4.4), we see that $N_B^s = O$. So $B = C_B + N_B$ is the core-nilpotent decomposition of B and $\mathrm{ind}(B) = s$. Since $\mathcal{R}(B^s) = \mathcal{R}(C_B)$ and $\mathcal{N}(B^s) = \mathcal{N}(C_B)$, we conclude that $\mathcal{R}(B^s) \cap \mathcal{N}(A^r) = \{0\}$ and $\mathcal{N}(B^s) \cap \mathcal{R}(A^r) = \{0\}$. Thus $B \in (\mathcal{C}_s)$ and $\mathrm{ind}(B) = s$. ∎

Remark 4.2. Conditions (b) and (d) in Theorem 4.1 can be replaced by the corresponding symmetrical conditions, as expressed in Remark 3.2.

Corollary 4.3. Let $A \in \mathbb{C}^{n\times n}$, $\mathrm{ind}(A) = r$, and let $B \in \mathbb{C}^{n\times n}$ be such that $\mathrm{ind}(B) = s$. Then the following are equivalent:

(a) $B \in (\mathcal{C}_s)$.
(b) $I + (A^{\mathrm{D}})^s(B^s - A^s)$ is nonsingular and $B^s(I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi = O$.
(c) $\mathrm{rank}\, B^s = \mathrm{rank}\, A^r$ and $I + (A^{\mathrm{D}})^s(B^s - A^s)$ is nonsingular.

Proof. (a)⇔(b). This equivalence follows from the equivalence (a)⇔(b) established in Theorem 4.1 if we show that, under assumption $\mathrm{ind}(B) = s$, the condition (b) in this theorem implies that $I + (A^{\mathrm{D}})^{s+1}(B^{s+1} - A^{s+1})$ is nonsingular. First, we observe that $\mathcal{N}(B^s) = \mathcal{N}\left(B^{s+1}\right)$ because $\mathrm{ind}(B) = s$. Now, since $A^\pi + (A^{\mathrm{D}})^s B^s$ is nonsingular, then $\mathcal{N}(A^\pi) \cap \mathcal{N}(B^s) = \{0\}$. From $B^s(I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}A^\pi = O$ it follows that $B^s = B^s(I + (A^{\mathrm{D}})^s(B^s - A^s))^{-1}(A^{\mathrm{D}})^s B^s$. So, we see that $\mathcal{N}\left((A^{\mathrm{D}})^{s+1}B^{s+1}\right) = \mathcal{N}\left((A^{\mathrm{D}})^s B^{s+1}\right) \subseteq \mathcal{N}\left(B^{s+1}\right)$. Thus $A^\pi + (A^{\mathrm{D}})^{s+1}B^{s+1}$ is nonsingular because $\mathcal{N}(A^\pi) \cap \mathcal{N}\left(B^{s+1}\right) = \{0\}$.

(a)⇔(c). This equivalence follows from the equivalence (a)⇔(d) established in Theorem 4.1. The details are omitted. ∎

Next, we give a representation for the powers of B.

Lemma 4.4. Let $A \in \mathbb{C}^{n\times n}$, $\mathrm{ind}(A) = r$, let $B \in \mathbb{C}^{n\times n}$, $\mathrm{ind}(B) = s$, and assume that $B \in (\mathcal{C}_s)$. Then, for $k \geq 1$,

$$B^k = P\left\{\begin{bmatrix} I \\ S \end{bmatrix} (B_1(I+TS))^{k-1}B_1 \begin{bmatrix} I & T \end{bmatrix} + \begin{bmatrix} T \\ -I \end{bmatrix} (B_2(I+ST))^{k-1}B_2 \begin{bmatrix} S & -I \end{bmatrix}\right\} P^{-1},$$

where $B_1 = I + TS$ is nonsingular and $B_2(I+ST)$ is nilpotent of index s.

Proof. The formula for the powers B^k can be derived from the representation (4.1), using the formula (1.3) of Lemma 1.1 and the formula (4.4). ∎

5. Perturbation results. In this section we give an explicit representation of B^{D} and we derive perturbation bounds of the Drazin inverse and the eigenprojection at zero.

Theorem 5.1. Let $A \in \mathbb{C}^{n\times n}$, $\mathrm{ind}(A) = r > 0$, let $B \in \mathbb{C}^{n\times n}$, $\mathrm{ind}(B) = s$, and assume that $B \in (\mathcal{C}_s)$. Let $E_1 = E = B - A$ and $E_s = B^s - A^s$. If $I + A^{\mathrm{D}}E$ is nonsingular, then

(5.1)
$$B^{\mathrm{D}} = \Phi_1^{-1}\Big(A^{\mathrm{D}} + A^{\mathrm{D}}\Psi_{ss}^{-1}\Phi_s^{-1}(A^{\mathrm{D}})^s E_s A^\pi (I - A^\pi E_s (A^{\mathrm{D}})^s \widetilde{\Phi}_s^{-1}) + A^\pi E_s (A^{\mathrm{D}})^s \widetilde{\Phi}_s^{-1}\Psi_{1s}^{-1}$$
$$\times \left(A^{\mathrm{D}} - \Phi_1^{-1}A^{\mathrm{D}}EA^{\mathrm{D}} - \Phi_1^{-1}A^{\mathrm{D}}(\Psi_{ss} - I)\Psi_{ss}^{-1}\right)(I + \Phi_s^{-1}(A^{\mathrm{D}})^s E_s A^\pi)\Big),$$

... $\Phi_i = I + (A^{\mathrm{D}})^i E_i$, $\widetilde{\Phi}_i = I + E_i (A^{\mathrm{D}})^i$, ... $\Psi_{is} = I + \Phi_i^{-1} (A^{\mathrm{D}})^i E_i A^{\pi} E_s (A^{\mathrm{D}})^s \widetilde{\Phi}_s^{-1}$ ... $i = 1$ ... $i = s$ ... $\max\{\|A^{\mathrm{D}}E\|, \|(A^{\mathrm{D}})^s E_s\|, \|E_s (A^{\mathrm{D}})^s\|\} < 1$ ...

(5.2)

$$\begin{aligned}
&\frac{\|B^{\mathrm{D}} - A^{\mathrm{D}}\|}{\|A^{\mathrm{D}}\|} \\
&\quad \le \frac{\|A^{\mathrm{D}}E\|}{1 - \|A^{\mathrm{D}}E\|} + \frac{\|(A^{\mathrm{D}})^s E_s A^{\pi}\| \|\Psi_{ss}^{-1}\|}{(1 - \|A^{\mathrm{D}}E\|)(1 - \|(A^{\mathrm{D}})^s E_s\|)} \left(1 + \frac{\|A^{\pi} E_s (A^{\mathrm{D}})^s\|}{1 - \|E_s (A^{\mathrm{D}})^s\|}\right) \\
&\qquad + \frac{\|A^{\pi} E_s (A^{\mathrm{D}})^s\| \|\Psi_{1s}^{-1}\|}{(1 - \|A^{\mathrm{D}}E\|)(1 - \|E_s (A^{\mathrm{D}})^s\|)} \left(1 + \frac{\|(A^{\mathrm{D}})^s E_s A^{\pi}\|}{1 - \|(A^{\mathrm{D}})^s E_s\|}\right) \\
&\qquad \times \left(1 + \frac{\|A^{\mathrm{D}}E\|}{1 - \|A^{\mathrm{D}}E\|} + \frac{\|(A^{\mathrm{D}})^s E_s A^{\pi}\| \|A^{\pi} E_s (A^{\mathrm{D}})^s\| \|\Psi_{ss}^{-1}\|}{(1 - \|A^{\mathrm{D}}E\|)(1 - \|E_s (A^{\mathrm{D}})^s\|)(1 - \|(A^{\mathrm{D}})^s E_s\|)}\right).
\end{aligned}$$

...

$$\max\{\|A^{\mathrm{D}}E\|, \|(A^{\mathrm{D}})^s E_s\|, \|E_s (A^{\mathrm{D}})^s\|\} < \frac{1}{1 + \sqrt{\|A^{\pi}\|}},$$

... $i = 1$ ... $i = s$

$$\|\Psi_{is}^{-1}\| \le \frac{(1 - \|(A^{\mathrm{D}})^i E_i\|)(1 - \|E_s (A^{\mathrm{D}})^s\|)}{(1 - \|(A^{\mathrm{D}})^i E_i\|)(1 - \|E_s (A^{\mathrm{D}})^s\|) - \|(A^{\mathrm{D}})^i E_i\| \|A^{\pi} E_s (A^{\mathrm{D}})^s\|}. \tag{5.3}$$

... From Theorem 4.1(c), we have that the index 1-nilpotent decomposition of B is given by $B = C_B + N_B$, with $C_B = P \begin{bmatrix} I \\ S \end{bmatrix} B_1 \begin{bmatrix} I & T \end{bmatrix} P^{-1}$ and $N_B = P \begin{bmatrix} T \\ -I \end{bmatrix} B_2 \begin{bmatrix} S & -I \end{bmatrix} P^{-1}$, where B_1 and $I + TS$ are nonsingular and $B_2(I + ST)$ is nilpotent of index s. Hence, applying Lemma 3.3, formulae (3.3), we obtain

$$B^{\mathrm{D}} = C_B^{\sharp} = P \begin{bmatrix} I \\ S \end{bmatrix} [(I + TS) B_1 (I + TS)]^{-1} \begin{bmatrix} I & T \end{bmatrix} P^{-1}. \tag{5.4}$$

Furthermore, we can write $E = B - A$ as

$$E = P \begin{pmatrix} B_1 + TB_2S - A_1 & B_1T - TB_2 \\ SB_1 - B_2S & SB_1T + B_2 - A_2 \end{pmatrix} P^{-1}. \tag{5.5}$$

In view of this latter representation we get

$$I + A^{\mathrm{D}}E = P \begin{pmatrix} A_1^{-1}(B_1 + TB_2S) & A_1^{-1}(B_1T - TB_2) \\ O & I \end{pmatrix} P^{-1}. \tag{5.6}$$

From the assumption that $I + A^{\mathrm{D}}E$ is nonsingular, it follows that $B_1 + TB_2S$ is nonsingular. Using (5.6) and (5.4) we obtain

(5.7)

$$(I + A^{\mathrm{D}}E) B^{\mathrm{D}} = P \begin{pmatrix} A_1^{-1}(I + TS)^{-1} & A_1^{-1}(I + TS)^{-1}T \\ S((I + TS)B_1(I + TS))^{-1} & S((I + TS)B_1(I + TS))^{-1}T \end{pmatrix} P^{-1}.$$

By denoting $\Phi_1 = I + A^{\mathrm{D}}E$, in view of (5.6) we obtain

$$\Phi_1^{-1} = P\begin{pmatrix} (B_1+TB_2S)^{-1}A_1 & -(B_1+TB_2S)^{-1}(B_1T-TB_2) \\ O & I \end{pmatrix} P^{-1}. \tag{5.8}$$

Utilizing the representations of the powers of B given in Lemma 4.4, we write $E_s = B^s - A^s$ as

$$E_s = P\begin{pmatrix} (B_1(I+TS))^{s-1}B_1 - A_1^s & (B_1(I+TS))^{s-1}B_1T \\ S(B_1(I+TS))^{s-1}B_1 & S(B_1(I+TS))^{s-1}B_1T - A_2^s \end{pmatrix} P^{-1}\ .$$

By denoting $\Phi_s = I + (A^{\mathrm{D}})^s E_s$ and $\widetilde{\Phi}_s = I + E_s(A^{\mathrm{D}})^s$ we get

$$\begin{aligned} \Phi_s^{-1} &= P\begin{pmatrix} B_1^{-1}\big((B_1(I+TS))^{(s-1)}\big)^{-1}A_1^s & -T \\ O & I \end{pmatrix} P^{-1}, \\ \widetilde{\Phi}_s^{-1} &= P\begin{pmatrix} A_1^sB_1^{-1}\big((B_1(I+TS))^{(s-1)}\big)^{-1} & O \\ -S & I \end{pmatrix} P^{-1}, \end{aligned} \tag{5.9}$$

and, hence,

$$\Phi_s^{-1}(A^{\mathrm{D}})^s = (A^{\mathrm{D}})^s\widetilde{\Phi}_s^{-1} = P\begin{pmatrix} B_1^{-1}\big((B_1(I+TS))^{(s-1)}\big)^{-1} & O \\ O & O \end{pmatrix} P^{-1}.$$

Furthermore,

$$\Phi_s^{-1}(A^{\mathrm{D}})^s E_s A^{\pi} = P\begin{pmatrix} O & T \\ O & O \end{pmatrix} P^{-1},\ \ A^{\pi}E_s(A^{\mathrm{D}})^s\widetilde{\Phi}_s^{-1} = P\begin{pmatrix} O & O \\ S & O \end{pmatrix} P^{-1}. \tag{5.10}$$

Let $\Psi_{is} = I + \Phi_i^{-1}(A^{\mathrm{D}})^i E_i A^{\pi} E_s (A^{\mathrm{D}})^s \widetilde{\Phi}_s^{-1}$ for $i = 1$ and $i = s$. Using (5.10) we see that

$$\Psi_{ss}^{-1} = P\begin{pmatrix} (I+TS)^{-1} & O \\ O & I \end{pmatrix} P^{-1}, \tag{5.11}$$

and, using (5.5), (5.6), and (5.10) we obtain

$$\begin{aligned} \Psi_{1s} &= P\left[\begin{pmatrix} I & O \\ O & I \end{pmatrix}\right. \\ &\qquad \left. + \begin{pmatrix} I-(B_1+TB_2S)^{-1}A_1 & (B_1+TB_2S)^{-1}(B_1T-TB_2) \\ O & O \end{pmatrix}\begin{pmatrix} O & O \\ S & O \end{pmatrix}\right] P^{-1} \\ &= P\begin{pmatrix} (B_1+TB_2S)^{-1}B_1(I+TS) & O \\ O & I \end{pmatrix} P^{-1}, \end{aligned}$$

and, thus,

$$\Psi_{1s}^{-1} = P\begin{pmatrix} (I+TS)^{-1}B_1^{-1}(B_1+TB_2S) & O \\ O & I \end{pmatrix} P^{-1}. \tag{5.12}$$

Now, let us introduce

$$\begin{aligned}\Sigma_1 &= A^{\mathrm{D}} + A^{\mathrm{D}}\Psi_{ss}^{-1}\Phi_s^{-1}(A^{\mathrm{D}})^s E_s A^{\pi}(I - A^{\pi}E_s(A^{\mathrm{D}})^s\widetilde{\Phi}_s^{-1}),\\ \Omega &= A^{\mathrm{D}} - \Phi_1^{-1}A^{\mathrm{D}}EA^{\mathrm{D}} - \Phi_1^{-1}A^{\mathrm{D}}(\Psi_{ss} - I)\Psi_{ss}^{-1},\\ \Sigma_2 &= A^{\pi}E_s(A^{\mathrm{D}})^s\widetilde{\Phi}_s^{-1}\Psi_{1s}^{-1}\Omega(I + \Phi_s^{-1}(A^{\mathrm{D}})^s E_s A^{\pi}).\end{aligned} \tag{5.13}$$

In order to verify identity (5.1) we will see that the matrix representation of $\Sigma_1 + \Sigma_2$ is equal to the right-hand side of (5.7). We compute

$$\begin{aligned}\Sigma_1 &= P\left[\begin{pmatrix} A_1^{-1} & O\\ O & O\end{pmatrix} + \begin{pmatrix} O & A_1^{-1}(I+TS)^{-1}T\\ O & O\end{pmatrix}\begin{pmatrix} I & O\\ -S & I\end{pmatrix}\right]P^{-1}\\ &= P\begin{pmatrix} A_1^{-1}(I+TS)^{-1} & A_1^{-1}(I+TS)^{-1}T\\ O & O\end{pmatrix}P^{-1}.\end{aligned}$$

On the other hand, utilizing (5.5), (5.8), and (5.11) we see that

$$\Omega = P\begin{pmatrix}(B_1 + TB_2S)^{-1}(I+TS)^{-1} & O\\ O & O\end{pmatrix}P^{-1},$$

and, hence, using (5.12), we get

$$\Psi_{1s}^{-1}\Omega = P\begin{pmatrix}(I+TS)^{-1}B_1^{-1}(I+TS)^{-1} & O\\ O & O\end{pmatrix}P^{-1}.$$

Therefore,

$$\begin{aligned}\Sigma_2 =& P\begin{pmatrix} O & O\\ S & O\end{pmatrix}\begin{pmatrix}(I+TS)^{-1}B_1^{-1}(I+TS)^{-1} & O\\ O & I\end{pmatrix}\begin{pmatrix} I & T\\ O & I\end{pmatrix}P^{-1}\\ =& P\begin{pmatrix} O & O\\ S(I+TS)^{-1}B_1^{-1}(I+TS)^{-1} & S(I+TS)^{-1}B_1^{-1}(I+TS)^{-1}T\end{pmatrix}P^{-1}.\end{aligned}$$

In view of these expressions of Σ_1 and Σ_2 we conclude the proof of the first part. From the identity $B^{\mathrm{D}} - A^{\mathrm{D}} + A^{\mathrm{D}}E(B^{\mathrm{D}} - A^{\mathrm{D}} + A^{\mathrm{D}}) = \Sigma_1 - A^{\mathrm{D}} + \Sigma_2$, taking norms we obtain

$$\|B^{\mathrm{D}} - A^{\mathrm{D}}\| \le \|A^{\mathrm{D}}E\|\|B^{\mathrm{D}} - A^{\mathrm{D}}\| + \|A^{\mathrm{D}}E\|\|A^{\mathrm{D}}\| + \|\Sigma_1 - A^{\mathrm{D}}\| + \|\Sigma_2\|.$$

Since $\max\{\|A^{\mathrm{D}}E\|, \|(A^{\mathrm{D}})^sE_s\|, \|E_s(A^{\mathrm{D}})^s\|\} < 1$, we have

$$\|B^{\mathrm{D}} - A^{\mathrm{D}}\| \le \frac{\|A^{\mathrm{D}}\|\|A^{\mathrm{D}}E\| + \|\Sigma_1 - A^{\mathrm{D}}\| + \|\Sigma_2\|}{1 - \|A^{\mathrm{D}}E\|} \tag{5.14}$$

and

$$\|\Phi_s^{-1}\| \le \frac{1}{1 - \|(A^{\mathrm{D}})^sE_s\|} \quad \text{and} \quad \|\widetilde{\Phi}_s^{-1}\| \le \frac{1}{1 - \|E_s(A^{\mathrm{D}})^s\|}. \tag{5.15}$$

Taking norms in (5.13), and using these upper bounds, we get

$$\|\Sigma_1 - A^{\mathrm{D}}\| \le \frac{\|A^{\mathrm{D}}\|\|(A^{\mathrm{D}})^sE_sA^{\pi}\|\|\Psi_{ss}^{-1}\|}{1 - \|(A^{\mathrm{D}})^sE_s\|}\left(1 + \frac{\|A^{\pi}E_s(A^{\mathrm{D}})^s\|}{1 - \|E_s(A^{\mathrm{D}})^s\|}\right)$$

and

$$\|\Sigma_2\| \leq \frac{\|A^{\mathrm{D}}\|\|A^{\pi}E_s(A^{\mathrm{D}})^s\|\|\Psi_{1s}^{-1}\|}{1-\|E_s(A^{\mathrm{D}})^s\|}\left(1+\frac{\|(A^{\mathrm{D}})^sE_sA^{\pi}\|}{1-\|(A^{\mathrm{D}})^sE_s\|}\right)$$
$$\times\left(1+\frac{\|A^{\mathrm{D}}E\|}{1-\|A^{\mathrm{D}}E\|}+\frac{\|(A^{\mathrm{D}})^sE_sA^{\pi}\|\|A^{\pi}E_s(A^{\mathrm{D}})^s\|\|\Psi_{ss}^{-1}\|}{(1-\|A^{\mathrm{D}}E\|)(1-\|E_s(A^{\mathrm{D}})^s\|)(1-\|(A^{\mathrm{D}})^sE_s\|)}\right).$$

Substituting these upper bounds of $\|\Sigma_1 - A^{\mathrm{D}}\|$ and $\|\Sigma_2\|$ in (5.14) we conclude the proof of (5.2). Finally, if $\max\{\|A^{\mathrm{D}}E\|, \|(A^{\mathrm{D}})^sE_s\|, \|E_s(A^{\mathrm{D}})^s\|\} < \frac{1}{1+\sqrt{\|A^{\pi}\|}}$, then

$$\|\Psi_{is} - I\| \leq \frac{\|(A^{\mathrm{D}})^iE_i\|\|A^{\pi}E_s(A^{\mathrm{D}})^s\|}{(1-\|(A^{\mathrm{D}})^iE_i\|)(1-\|E_s(A^{\mathrm{D}})^s\|)} < 1, \quad i = 1, s.$$

Hence, it follows that

$$\|\Psi_{is}^{-1}\| \leq \frac{(1-\|(A^{\mathrm{D}})^iE_i\|)(1-\|E_s(A^{\mathrm{D}})^s\|)}{(1-\|(A^{\mathrm{D}})^iE_i\|)(1-\|E_s(A^{\mathrm{D}})^s\|)-\|(A^{\mathrm{D}})^iE_i\|\|A^{\pi}E_s(A^{\mathrm{D}})^s\|}, \quad i = 1, s.$$

This completes the proof. □

. . . 5.2. If we denote $\delta_{is} = (1-\|(A^{\mathrm{D}})^iE_i\|)(1-\|E_s(A^{\mathrm{D}})^s\|) - \|(A^{\mathrm{D}})^iE_i\|$ $\|A^{\pi}E_s(A^{\mathrm{D}})^s\|$, then the upper bounds (5.3), for $i = 1$ and $i = s$, can be expressed as

$$\|\Psi_{is}^{-1}\| \leq 1 + \frac{\|(A^{\mathrm{D}})^iE_i\|\|A^{\pi}E_s(A^{\mathrm{D}})^s\|}{\delta_{is}} = 1 + O(\|E\|^2),$$

where in the last identity we have taken into account that $\|E_s\| = O(\|E\|)$ (see [11]).

Substituting this in (5.2) we get that the upper bound of $\|B^{\mathrm{D}} - A^{\mathrm{D}}\|$ up to the first order of $\|E\|$, has the following expression

$$\frac{\|B^{\mathrm{D}}-A^{\mathrm{D}}\|}{\|A^{\mathrm{D}}\|} \leq \frac{\|A^{\mathrm{D}}E\|}{1-\|A^{\mathrm{D}}E\|}+\frac{\|(A^{\mathrm{D}})^sE_sA^{\pi}\|}{(1-\|A^{\mathrm{D}}E\|)(1-\|(A^{\mathrm{D}})^sE_s\|)}$$
$$+\frac{\|A^{\pi}E_s(A^{\mathrm{D}})^s\|}{(1-\|A^{\mathrm{D}}E\|)(1-\|E_s(A^{\mathrm{D}})^s\|)}+O(\|E\|^2). \tag{5.16}$$

In the following corollary we show that the matrices satisfying condition (1.1), or equivalently $B^{\pi} = A^{\pi}$, are a particular case of the matrices satisfying condition $(\mathcal{C}_s)$.

Corollary 5.3. . . $A \in \mathbb{C}^{n\times n}$, $\mathrm{ind}(A) = r > 0$, . . . $B \in \mathbb{C}^{n\times n}$ $\mathrm{ind}(B) = s$ . . . . . $(\mathcal{C}_s)$ . . . $E = B - A$. $A^{\pi}EA^{\mathrm{D}} = A^{\mathrm{D}}EA^{\pi}$ . . . . . $B^{\mathrm{D}} = (I + A^{\mathrm{D}}E)^{-1}A^{\mathrm{D}}$. . . . . $\|A^{\mathrm{D}}E\| < 1$ . . .

$$\frac{\|B^{\mathrm{D}}-A^{\mathrm{D}}\|}{\|A^{\mathrm{D}}\|} \leq \frac{\|A^{\mathrm{D}}E\|}{1-\|A^{\mathrm{D}}E\|}. \tag{5.17}$$

. . . . We have that E has the representation (5.5) given in the proof of Theorem 5.1. From condition $A^{\pi}EA^{\mathrm{D}} = A^{\mathrm{D}}EA^{\pi}$ it follows that

$$B_1T = TB_2 \text{ and } SB_1 = B_2S.$$

Using these relations we get that

$$S(B_1(I+TS))^s = B_2(I+ST)S(B_1(I+TS))^{s-1} = \cdots = (B_2(I+ST))^sS.$$

Applying that $B_2(I+ST)$ is nilpotent of index s and $B_1(I+TS)$ is nonsingular we obtain that $S=O$. Analogously, we can see that $T=O$. Thus, expression (5.6) takes the form

$$I+A^{\mathrm{D}}E=P\begin{pmatrix}A_1^{-1}B_1 & O\\ O & I\end{pmatrix}P^{-1}.$$

Clearly $I+A^{\mathrm{D}}E$ is nonsingular. In view of (5.4) we get

$$B^{\mathrm{D}}=P\begin{pmatrix}B_1^{-1} & O\\ O & O\end{pmatrix}P^{-1}=(I+A^{\mathrm{D}}E)^{-1}A^{\mathrm{D}}.$$

Hence, we get that $B^{\pi}=A^{\pi}$ and the upper bound (5.17). □

THEOREM 5.4. [illegible] $A\in\mathbb{C}^{n\times n}$, $\mathrm{ind}(A)=r>0$, [illegible] $B\in\mathbb{C}^{n\times n}$, $\mathrm{ind}(B)=s$, [illegible] $(\mathcal{C}_s)$ [illegible] $E_s=B^s-A^s$, $\max\{\|(A^{\mathrm{D}})^sE_s\|,\|E_s(A^{\mathrm{D}})^s\|\}<1$, [illegible]

$$\|B^{\pi}-A^{\pi}\|\le\frac{\|(A^{\mathrm{D}})^sE_sA^{\pi}\|}{1-\|(A^{\mathrm{D}})^sE_s\|}+\frac{\|A^{\pi}E_s(A^{\mathrm{D}})^s\|\|\Psi_{ss}^{-1}\|}{(1-\|(A^{\mathrm{D}})^sE_s\|)(1-\|E_s(A^{\mathrm{D}})^s\|)}\left(1+\frac{\|(A^{\mathrm{D}})^sE_sA^{\pi}\|}{1-\|(A^{\mathrm{D}})^sE_s\|}\right),\tag{5.18}$$

[illegible] $\Psi_{ss}=I+(I+(A^{\mathrm{D}})^sE_s)^{-1}(A^{\mathrm{D}})^sE_sA^{\pi}E_s(A^{\mathrm{D}})^s(I+E_s(A^{\mathrm{D}})^s)^{-1}$ [illegible] $\max\{\|(A^{\mathrm{D}})^sE_s\|,\|E_s(A^{\mathrm{D}})^s\|\}<\frac{1}{1+\sqrt{\|A^{\pi}\|}}$ [illegible] $\|\Psi_{ss}^{-1}\|$ [illegible] (5.3).

[illegible]. From Theorem 2.3 we have

$$B^{\pi}+(A^{\mathrm{D}})^sE_sB^{\pi}=-A^{\pi}X^{-1},\tag{5.19}$$

where $X=I-(I+(A^{\mathrm{D}})^sE_s)^{-1}A^{\pi}-A^{\pi}(I+E_s(A^{\mathrm{D}})^s)^{-1}$. Utilizing the expressions of Φ_s^{-1} and $\widetilde{\Phi}_s^{-1}$ given in the proof of Theorem 5.1 by (5.9), we can represent

$$X=P\begin{pmatrix}I & T\\ S & -I\end{pmatrix}P^{-1}\ \text{ and }\ X^{-1}=P\begin{pmatrix}(I+TS)^{-1} & (I+TS)^{-1}T\\ S(I+TS)^{-1} & -I+S(I+TS)^{-1}T\end{pmatrix}P^{-1}.$$

Thus,

$$-A^{\pi}X^{-1}=A^{\pi}+P\begin{pmatrix}O & O\\ -S(I+TS)^{-1} & -S(I+TS)^{-1}T\end{pmatrix}P^{-1}.$$

Hence, in view of the representations (5.10) and (5.11) we may write

$$-A^{\pi}X^{-1}=A^{\pi}-A^{\pi}E_s(A^{\mathrm{D}})^s\widetilde{\Phi}_s^{-1}\Psi_{ss}^{-1}(I+\Phi_s^{-1}(A^{\mathrm{D}})^sE_sA^{\pi}).$$

Substituting the latter identity in (5.19) we obtain

$$B^{\pi}-A^{\pi}=-(A^{\mathrm{D}})^sE_s(B^{\pi}-A^{\pi}+A^{\pi})-A^{\pi}E_s(A^{\mathrm{D}})^s\widetilde{\Phi}_s^{-1}\Psi_{ss}^{-1}(I+\Phi_s^{-1}(A^{\mathrm{D}})^sE_sA^{\pi}).$$

Taking norms

$$\|B^{\pi}-A^{\pi}\|\le\|(A^{\mathrm{D}})^sE_s\|\|B^{\pi}-A^{\pi}\|+\|(A^{\mathrm{D}})^sE_sA^{\pi}\|+\|A^{\pi}E_s(A^{\mathrm{D}})^s\|\|\widetilde{\Phi}_s^{-1}\|\|\Psi_{ss}^{-1}\|(1+\|\Phi_s^{-1}\|\|(A^{\mathrm{D}})^sE_sA^{\pi}\|).$$

TABLE 5.1
Comparison of upper bounds of $\|BB^{\mathrm{D}} - AA^{\mathrm{D}}\|_2$.

	Exact value	[13, Thm. 5], (15)	(5.18)
$B = A + E_1$	9.99×10^{-10}	1.00×10^{-5}	1.00×10^{-9}
$B = A + E_2$	1.85×10^{-9}	2.74×10^{-5}	2.74×10^{-9}

TABLE 5.2
Comparison of upper bounds of $\|B^{\mathrm{D}} - A^{\mathrm{D}}\|_2/\|A^{\mathrm{D}}\|_2$.

	$B = A + E_1$	$B = A + E_2$
Exact Value	1.12×10^{-10}	3.44×10^{-11}
[13, Thm. 1], (1)	0.7649	0.9008
[13, Thm. 4], (6)	$1.00 \times 10^{-5} + O(\|E\|^2)$	$2.73 \times 10^{-5} + O(\|E\|^2)$
(5.20)+(5.18)	3.41×10^{-9}	6.88×10^{-9}
(5.2)	2.41×10^{-9}	4.15×10^{-9}
(5.16)	$2.41 \times 10^{-9} + O(\|E\|^2)$	$4.15 \times 10^{-9} + O(\|E\|^2)$

TABLE 5.3
Comparison of upper bounds of $\|B^{\mathrm{D}} - A^{\mathrm{D}}\|_F/\|A^{\mathrm{D}}\|_F$.

	Exact value	[14, Thm. 4.1], (4.1)	(5.2)
$B = A + E_1$	1.14×10^{-10}	8.39×10^{-5}	2.42×10^{-9}
$B = A + E_2$	3.47×10^{-11}	8.39×10^{-5}	4.15×10^{-9}

Since $\max\{\|(A^{\mathrm{D}})^s E_s\|, \|E_s(A^{\mathrm{D}})^s\|\} < 1$, regrouping in $\|B^\pi - A^\pi\|$ and substituting $\|\Phi_s^{-1}\|$ and $\|\widetilde{\Phi}_s^{-1}\|$ by the upper bounds (5.15), we get (5.18). □

5.5. If $\max\{\|A^{\mathrm{D}}E\|, \|(A^{\mathrm{D}})^s E_s\|, \|E_s(A^{\mathrm{D}})^s\|\} < \frac{1}{1+\sqrt{\|A^\pi\|}}$, as we have seen in Remark 5.2, the upper bound of $\|B^\pi - A^\pi\|$ up to the first order of $\|E\|$ has the following expression:

$$\|B^\pi - A^\pi\| \leq \frac{\|(A^{\mathrm{D}})^s E_s A^\pi\|}{1 - \|(A^{\mathrm{D}})^s E_s\|} + \frac{\|A^\pi E_s (A^{\mathrm{D}})^s\|}{(1 - \|(A^{\mathrm{D}})^s E_s\|)(1 - \|E_s(A^{\mathrm{D}})^s\|)} + O(\|E\|^2).$$

5.6. In [5, Theorem 3.1 and Remark 3.3], under assumption $\Delta + \|A^{\mathrm{D}}E\| < 1$, where Δ is un upper bound of $\|B^\pi - A^\pi\|$, the following estimation of the Drazin inverse was given:

$$\frac{\|B^{\mathrm{D}} - A^{\mathrm{D}}\|}{\|A^{\mathrm{D}}\|} \leq \frac{\|A^{\mathrm{D}}E\| + 2\Delta}{1 - \|A^{\mathrm{D}}E\| - \Delta}. \tag{5.20}$$

5.7. In Table 5.1 we compare the upper bound for $\|B^\pi - A^\pi\|_2$ derived in Theorem 5.4 with the upper bound given in [13, Theorem 5]. The upper bounds for $\|B^{\mathrm{D}} - A^{\mathrm{D}}\|_2/\|A^{\mathrm{D}}\|_2$ given in Theorem 5.1, Remark 5.2, and Remark 5.6, replacing Δ in (5.20) by the upper bound given in (5.18), are compared in Table 5.2 with the upper bounds given in [13]. Let

$$A = \begin{pmatrix} \frac{1}{100} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \; E_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \epsilon & 0 & \epsilon & 0 \\ 0 & 0 & 0 & \epsilon & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \; E_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \epsilon & \epsilon & 0 \\ 0 & 0 & 0 & \epsilon & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where $\epsilon = 10^{-9}$. We have $\mathrm{ind}(A) = \mathrm{ind}(A + E_i) = 2$ and $\mathrm{rank}\, A^2 = \mathrm{rank}(A + E_i)^2 = \mathrm{rank}\, A^2(A + E_i)^2A^2 = 3$, $i = 1, 2$. By Theorem 2.1 we have that $B = A + E_i$ satisfies condition $(\mathcal{C}_2)$.

In Table 5.3 we compare the upper bound (5.2) using the Frobenius norm with the upper bound given in [14], formula (4.1). That formula is based on the separation of matrices $\mathrm{sep}_F(C, N)$, with C and N being the matrices in the following Schur decomposition,

$$Q^H AQ = \begin{bmatrix} C & G \\ O & N \end{bmatrix},$$

where Q is an unitary matrix, C is nonsingular, and N is nilpotent of index $\mathrm{ind}(A)$. In this example $\mathrm{sep}_F(C, N) = 1.42 \times 10^{-4}$.

REFERENCES

[1] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses. Theory and Applications*, 2nd ed., Springer-Verlag, New York, 2003.

[2] S. L. Campbell and C. D. Meyer, Jr., *Continuity properties of the Drazin pseudoinverse*, Linear Algebra Appl., 10 (1975), pp. 77–83.

[3] S. L. Campbell and C. D. Meyer, *Generalized Inverses of Linear Transformations*, Dover, New York, 1991 (corrected reprint of the original published by Pitman, London, 1979).

[4] N. Castro-González, J. J. Koliha, and Y. Wei, *Perturbation of the Drazin inverse for matrices with equal eigenprojection at zero*, Linear Algebra Appl., 312 (2000), pp. 181–189.

[5] N. Castro-González and J. Y. Vélez-Cerrada, *Characterizations of matrices which eigenprojections at zero are equal to a fixed perturbation*, Appl. Math. Comput., 159 (2004), pp. 613–623.

[6] J. J. Koliha, *Error bound for a general perturbation of the Drazin inverse*, Appl. Math. Comput., 126 (2002), pp. 181–185.

[7] J. J. Koliha, V. Rakočević, and I. Straškraba, *The difference and sum of projectors*, Linear Algebra Appl., 388 (2004), pp. 279–288.

[8] X. Li and Y. Wei, *An improvement on the perturbation of the group inverse and oblique projection*, Linear Algebra Appl., 338 (2001), pp. 53–66.

[9] X. Li and Y. Wei, *An expression of the Drazin inverse of a perturbed matrix*, Appl. Math. Comput., 153 (2004), pp. 187–198.

[10] C. D. Meyer, Jr., *The condition of a finite Markov chain and perturbation bounds for the limiting probabilities*, SIAM J. Alg. Disc. Meth., 1 (1980), pp. 273–283.

[11] G.-H. Rong, *The error bound of the perturbation of the Drazin inverse*, Linear Algebra Appl., 47 (1982), pp. 159–168.

[12] Y. Wei, *On the perturbation of the group inverse and oblique projection*, Appl. Math. Comput., 98 (1999), pp. 29–42.

[13] Y. Wei and X. Li, *An improvement on perturbation bounds for the Drazin inverse*, Numer. Linear Algebra Appl., 10 (2003), pp. 563–575.

[14] Y. Wei, X. Li, and F. Bu, *A perturbation bound of the Drazin inverse of a matrix by separation of simple invariant subspaces*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 72–81.

[15] Y. Wei and G. Wang, *The perturbation theory for the Drazin inverse and its applications*, Linear Algebra Appl., 258 (1997), pp. 179–186.

[16] Y. Wei and H. Wu, *On the perturbation of the Drazin inverse and oblique projection*, Appl. Math. Lett., 13 (2000), pp. 77–83.

[17] F. Zhang, *Matrix Theory. Basic Results and Techniques*, Springer-Verlag, New York, 1999.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 898–924

WEIGHTED POLAR DECOMPOSITION AND WGL PARTIAL ORDERING OF RECTANGULAR COMPLEX MATRICES*

HU YANG† AND HANYU LI‡

Abstract. In this paper, the unique weighted polar decomposition theorem for rectangular complex matrices is first proved based on introducing the weighted partial isometric matrices. Then the simultaneous weighted polar decomposition of two rectangular complex matrices is studied, for which two sufficient conditions and one criterion are proposed. In order to obtain further characteristics of the simultaneous weighted polar decomposition, a new partial ordering called WGL partial ordering is defined on the set of rectangular matrices. Some basic properties of this new partial ordering are derived. In addition, we also provide methods for computing the weighted polar decomposition and discuss error bounds for the approximate generalized positive semidefinite polar factor.

Key words. weighted singular value decomposition, weighted partial isometric matrix, weighted polar decomposition, simultaneous weighted polar decomposition, WGL partial ordering, error bound

AMS subject classifications. 15A09, 15A18, 15A45, 65F30, 65F35

DOI. 10.1137/070700917

1. Introduction.

Let $\mathbb{C}^{m\times n}$ be the set of $m\times n$ complex matrices and $\mathbb{C}_r^{m\times n}$ be the subset of $\mathbb{C}^{m\times n}$ consisting of matrices with rank r. Let $\mathbb{C}_{\geq}^{m}$ be the set of Hermitian positive semidefinite matrices of order m, and $\mathbb{C}_{>}^{m}$ be the subset of $\mathbb{C}_{\geq}^{m}$ consisting of Hermitian positive definite matrices. Let I_r denote the identity matrix of order r. Given $A\in\mathbb{C}^{m\times n}$, the symbols A^*, $r(A)$, $\lambda_1(A)$, $\mathrm{tr}(A)$, $R(A)$, $N(A)$, $\|A\|_F$, $\|A\|_2$, and $\|A\|$ will stand for the conjugate transpose, rank, the largest eigenvalue, trace, range, null space, Frobenius norm, spectral norm, and unitarily invariant norm of A, respectively. Furthermore, without specification, we always assume that $m>n>r$ and the given weight matrices $M\in\mathbb{C}_{>}^{m}, N\in\mathbb{C}_{>}^{n}$.

For an arbitrary matrix $A\in\mathbb{C}^{m\times n}$, there is a unique matrix $X\in\mathbb{C}^{n\times m}$ satisfying the following equations:

$$AXA=A,\quad XAX=X,\quad (MAX)^*=MAX,\quad (NXA)^*=NXA. \tag{1.1}$$

Matrix X is known as the weighted Moore–Penrose inverse of A and denoted by $X=A_{MN}^{+}$. The weighted Moore–Penrose inverse has many applications in weighted linear least squares problems, prediction theory, numerical analysis, and so on (see, e.g., [19, 24]). In particular, when $M=I_m$ and $N=I_n$, the matrix X is reduced to the Moore–Penrose inverse of A and denoted by $X=A^{+}$.

Given the weight matrices M and N, the weighted inner products in $\mathbb{C}^m$ and $\mathbb{C}^n$ may be defined as

$$(x,y)_M=y^*Mx,\ x,y\in\mathbb{C}^m,\ \text{and}\ (x,y)_N=y^*Nx,\ x,y\in\mathbb{C}^n, \tag{1.2}$$

and the weighted vector norms are defined as

$$\|x\|_M=(x^*Mx)^{1/2}=\left\|M^{1/2}x\right\|_2,\qquad x\in\mathbb{C}^m,$$

*Received by the editors August 24, 2007; accepted for publication (in revised form) by R.-C. Li April 11, 2008; published electronically September 17, 2008.
http://www.siam.org/journals/simax/30-2/70091.html

†College of Mathematics and Physics, Chongqing University, Chongqing, 400030, People's Republic of China (yh@cqu.edu.cn).

‡Corresponding author. College of Mathematics and Physics, Chongqing University, Chongqing, 400030, People's Republic of China (lihy.hy@gmail.com).

and

$$\|x\|_N = (x^*Nx)^{1/2} = \left\|N^{1/2}x\right\|_2, \qquad x \in \mathbb{C}^n.$$

Moreover, if $A \in \mathbb{C}^{m\times n}$, then from [19, 24], the matrix $X \in \mathbb{C}^{n\times m}$ satisfying

$$(Ax, y)_M = (x, Xy)_N \quad \text{for all } x \in \mathbb{C}^n,\ y \in \mathbb{C}^m$$

is called the [illegible] (or adjoint) of the matrix A and denoted by $X = A^{\#}$. This together with (1.2) implies

$$A^{\#} = N^{-1}A^*M. \tag{1.3}$$

We now recall two partial orderings and (generalized) polar decomposition of complex matrices as follows.

For $A, B \in \mathbb{C}^{m\times n}$, we say that A is below B with respect to the [illegible] [16] and write $A \overset{\#}{\leq} B$ whenever $A^{\#}A = A^{\#}B$ and $AA^{\#} = BA^{\#}$. For $A, B \in \mathbb{C}^{m\times m}$, we say that A is below B with respect to the [illegible] and write $A \leq_L B$ whenever $(B - A) \in \mathbb{C}^m_{\geq}$.

For $A \in \mathbb{C}^{m\times n}_r$, there are a partial isometric matrix $E \in \mathbb{C}^{m\times n}$ and two Hermitian positive semidefinite matrices $G \in \mathbb{C}^{m\times m}, H \in \mathbb{C}^{n\times n}$ such that

$$A = GE = EH. \tag{1.4}$$

This decomposition is called the [illegible] [2, 22] of A, and E and G, H are called the [illegible] and [illegible] of this decomposition, respectively. Usually, when $r = n$, decomposition (1.4) is called the [illegible] and, in this case, E and H are called the [illegible] and [illegible], respectively.

There have been many published works on both the partial orderings of matrices and the (generalized) polar decomposition (see, e.g., [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 21, 22, 25]). Among these works, the properties of the (generalized) polar decomposition involved in the partial orderings of matrices were studied by Hauke, Markiewicz, Groß, and Zhuang [6, 8, 18, 25], and numerical methods including direct and iterative methods for computing the (generalized) polar decomposition could be found in [4, 5, 9, 10, 11, 12, 15, 22].

After obtaining an [illegible] solution X of the unitary polar factor E using iterative methods, Sun [21] took $K = \frac{1}{2}(X^*A + A^*X)$ as an approximate solution of the positive definite polar factor H and presented the error estimates for the approximate positive definite polar factor K. Chen and Li [3] extended some results obtained by Sun [21] to the generalized polar decomposition, and derived some error bounds for the approximate positive semidefinite polar factor. In the present paper, we will consider the weighted polar decomposition (WPD), a generalization of the (generalized) polar decomposition.

The rest of this paper is organized as follows. Section 2 provides some preliminaries. Section 3 first proves the unique WPD theorem for rectangular complex matrices after introducing the weighted partial isometric matrices and then defines and studies the simultaneous weighted polar decomposition (SWPD) of two rectangular complex matrices. Two sufficient conditions and a criterion for SWPD are obtained. In section 4, we present a new partial ordering of matrices called WGL partial ordering,

and then we discuss its basic properties. Using this new partial ordering, we study the characteristics of SWPD further. The methods for computing WPD and the error bounds for the approximate generalized positive semidefinite (AGPSD) polar factor are derived in section 5 and section 6, respectively.

2. Preliminaries. First, we introduce some results on the weighted matrix norms. From [24], we know that the [illegible] may be defined as

$$\|A\|_{MN} = \max_{\|x\|_N=1} \|Ax\|_M, \qquad A \in \mathbb{C}^{m\times n},\ x \in \mathbb{C}^n, \tag{2.1}$$

from which it is easy to get the following relation:

$$\|A\|_{MN} = \left\|M^{1/2}AN^{-1/2}\right\|_2. \tag{2.2}$$

We now define two other weighted matrix norms called the [illegible] and the [illegible], respectively, as follows:

$$\|A\|_{F(MN)} = \left\|M^{1/2}AN^{-1/2}\right\|_F, \quad A \in \mathbb{C}^{m\times n}, \tag{2.3}$$

$$\|A\|_{(MN)} = \left\|M^{1/2}AN^{-1/2}\right\|, \qquad A \in \mathbb{C}^{m\times n}. \tag{2.4}$$

It is worth pointing out that the weighted unitary invariant norm defined in (2.4) is essentially equivalent to the (M, N)-invariant norm presented by Rao and Rao [20].

According to the properties of the Frobenius norm and (1.3), we have

$$\|A\|_{F(MN)} = \left(\mathrm{tr}((M^{1/2}AN^{-1/2})^*(M^{1/2}AN^{-1/2}))\right)^{1/2} = (\mathrm{tr}(A^{\#}A))^{1/2}. \tag{2.5}$$

Hence, the weighted Frobenius norm is similar to the Frobenius norm in form.

From Van Loan [23], we know that the (M, N) [illegible] of $A \in \mathbb{C}_r^{m\times n}$ are the elements of the set $\sigma_{MN}(A)$ defined by

$$\sigma_{MN}(A) = \left\{\sigma : \sigma \geq 0, \sigma \text{ is a stationary value of } \frac{\|Ax\|_M}{\|x\|_N}\right\}.$$

By using Lagrange multipliers, for every nonzero element of the set $\sigma_{MN}(A)$, we have

$$\sigma_i = \lambda_i^{1/2}(N^{-1}A^*MA) = \lambda_i^{1/2}(A^{\#}A), \quad i = 1, \ldots, r, \tag{2.6}$$

which together with (2.5) gives

$$\|A\|_{F(MN)}^2 = \sum_{i=1}^{r} \sigma_i^2. \tag{2.7}$$

Van Loan [23] also presented the following (M, N) [illegible] (MN-SVD), which is useful in this paper.

LEMMA 2.1. [illegible] $A \in \mathbb{C}_r^{m\times n}$ [illegible] $U \in \mathbb{C}^{m\times m}$ [illegible] $V \in \mathbb{C}^{n\times n}$ [illegible] $U^*MU = I_m$ [illegible] $V^*N^{-1}V = I_n$ [illegible]

$$A = U\begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix}V^*, \tag{2.8}$$

$\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$, $\sigma_i = \sqrt{\lambda_i}$ $\lambda_1 \geq \cdots \geq \lambda_r > 0$ $A^{\#}A = N^{-1}A^*MA$ $\sigma_1 \geq \cdots \geq \sigma_r > 0$ (M, N) A A

$$A^{+}_{MN} = N^{-1}V \begin{pmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^*M. \tag{2.9}$$

$U = (U_r, U_{m-r})$ $V = (V_r, V_{n-r})$ $U_r = [u_1, \ldots, u_r] \in \mathbb{C}^{m\times r}$ $V_r = [v_1, \ldots, v_r] \in \mathbb{C}^{n\times r}$

$$A = U_r\Sigma V_r^* \quad A^{+}_{MN} = N^{-1}V_r\Sigma^{-1}U_r^*M. \tag{2.10}$$

$$\|A\|_{MN} = \sigma_1 \quad \left\|A^{+}_{MN}\right\|_{NM} = \frac{1}{\sigma_r}. \tag{2.11}$$

The following four lemmas will be needed in this paper. Lemmas 2.2 and 2.3 can be found in [1] and [13], respectively, and Lemmas 2.4 and 2.5 can be found in [24].

LEMMA 2.2. $A, B \in \mathbb{C}^m_{\geq}$

$$A \leq_L B \quad \lambda_1(B^{+}A) \leq 1 \quad R(A) \subseteq R(B).$$

LEMMA 2.3. $A, B \in \mathbb{C}^{m\times m}$ $P \in \mathbb{C}^{m\times k}$

$A \leq_L B$, $P^*AP \leq_L P^*BP$.

LEMMA 2.4. $A \in \mathbb{C}^{m\times n}_r$ $P \in \mathbb{C}^{k\times m}$ $Q \in \mathbb{C}^{l\times n}$ $P^*P = I_m$ $QQ^* = I_n$

$$(PAQ^*)^{+} = QA^{+}P^*.$$

LEMMA 2.5. $A \in \mathbb{C}^{m\times n}_r$ A^{+}_{MN}

$$A^{+}_{MN} = N^{-1/2}(M^{1/2}AN^{-1/2})^{+}M^{1/2}.$$

3. Weighted partial isometric matrix and WPD. Now we give the definition of the weighted partial isometric matrix.

DEFINITION 3.1. $E \in \mathbb{C}^{m\times n}$

$$\|Ex\|_M = \|x\|_N \quad x \in R(E^{\#}),$$

E (M, N) ( ) $E \in \mathbb{C}^{n\times m}$

$$\|Ex\|_N = \|x\|_M \quad x \in R(E^{\#}),$$

E (N, M) $M = I_m$ $N = I_n$ E ( ) [2, 19, 22])

The following lemma from [19] can be used to determine the orthogonal projector in the weighted inner product.

LEMMA 3.2. [illegible] $\mathbb{C}^m$ [illegible] (1.2) [illegible] $P \in \mathbb{C}^{m\times m}$ [illegible]

(a) $P^2 = P$ [illegible] (b) $(MP)^* = MP$ ($P^{\#} = P$).

From this lemma and the properties of the weighted Moore–Penrose inverse [24], i.e.,

$$R(AA^+_{MN}) = R(A), \quad R(A^+_{MN}A) = N^{-1}R(A^*) = R(A^{\#}), \tag{3.1}$$

we can get that the orthogonal projectors on $R(A)$ and $R(A^{\#})$ in the weighted inner product (1.2) may be given by

$$P_{R(A)} = AA^+_{MN} \text{ and } P_{R(A^{\#})} = A^+_{MN}A. \tag{3.2}$$

Using results from [17], Definition 3.1, and Lemma 3.2, we can draw some equivalent characterizations of the weighted partial isometric matrices.

LEMMA 3.3. [illegible] $E \in \mathbb{C}^{m\times n}$ [illegible]

(a) E [illegible]

(b) $E^{\#}$ [illegible]

(c) $E^+_{MN} = E^{\#}$.

(d) $E^{\#}E$ [illegible] $E^{\#}E = P_{R(E^{\#})}$.

(e) $EE^{\#}$ [illegible] $EE^{\#} = P_{R(E)}$.

(f) $EE^{\#}E = E$.

(g) $E^{\#}EE^{\#} = E^{\#}$.

In order to prove the WPD theorem, we need to introduce the following theorem.

THEOREM 3.4. [illegible] $A \in \mathbb{C}^{m\times n}_r$ [illegible] A U_r [illegible] V_r [illegible] 2.1 [illegible]

(a) $U_r^*MU_r = V_r^*N^{-1}V_r = I_r$ [illegible] $E = U_rV_r^*$ [illegible];

(b) $EE^{\#} = P_{R(A)}$ [illegible] $E^{\#}E = P_{R(A^{\#})}$.

[illegible]. (a) The first part can be obtained from $U^*MU = I_m$, $V^*N^{-1}V = I_n$, and the definitions of U_r and V_r easily.

That $E = U_rV_r^*$ is an MN-WPI matrix can be obtained from Lemma 3.3 and the following result:

$$EE^{\#}E = U_rV_r^*(U_rV_r^*)^{\#}U_rV_r^* = U_rV_r^*N^{-1}(U_rV_r^*)^*MU_rV_r^* = U_rV_r^* = E.$$

(b) From (3.2), Lemma 2.1, and the fact that $E = U_rV_r^*$, we have

$$P_{R(A)} = AA^+_{MN} = U_r\Sigma V_r^*N^{-1}V_r\Sigma^{-1}U_r^*M = UU_r^*M,$$

$$EE^{\#} = U_rV_r^*(U_rV_r^*)^{\#} = U_rV_r^*N^{-1}(U_rV_r^*)^*M = UU_r^*M.$$

Therefore, $EE^{\#} = P_{R(A)}$.

The equality $E^{\#}E = P_{R(A^{\#})}$ can also be proved in a similar way. ∎

THEOREM 3.5. $A \in \mathbb{C}_r^{m\times n}$ A

$$A = GE = EH, \tag{3.3}$$

$E \in \mathbb{C}^{m\times n}$ $MG \in \mathbb{C}_{\geq}^{m}$ $NH \in \mathbb{C}_{\geq}^{n}$ G H $G^{\#} = G$, $H^{\#} = H$. Let the MN-SVD of A be as in (2.8). For any k, where $r \leq k \leq \min\{m, n\}$, similar to U_r and V_r in Lemma 2.1, we define U_k, V_k, and Σ_k as follows:

$$U_k = [u_1, \ldots, u_k] \in \mathbb{C}^{m\times k}, \quad V_k = [v_1, \ldots, v_k] \in \mathbb{C}^{n\times k}, \quad \Sigma_k = \left(\begin{array}{c|c} \Sigma & 0 \\ \hline 0 & 0 \end{array}\right) \in \mathbb{C}^{k\times k}.$$

It is easy to obtain $U_k^* M U_k = V_k^* N^{-1} V_k = I_k$; then (2.8) can be rewritten as

$$\begin{aligned} A &= U_k \Sigma_k V_k^* = U_k \Sigma_k U_k^* M U_k V_k^* = (U_k \Sigma_k U_k^* M)(U_k V_k^*) \\ &= U_k V_k^* N^{-1} V_k \Sigma_k V_k^* = (U_k V_k^*)(N^{-1} V_k \Sigma_k V_k^*), \end{aligned}$$

which together with Theorem 3.4 proves (3.3) with the MN-WPI matrix

$$E = U_k V_k^* \tag{3.4}$$

and the generalized positive semidefinite matrices

$$G = U_k \Sigma_k U_k^* M, \qquad H = N^{-1} V_k \Sigma_k V_k^*. \tag{3.5}$$

In fact,

$$MG = M U_k \Sigma_k U_k^* M \in \mathbb{C}_{\geq}^{m}, \qquad NH = V_k \Sigma_k V_k^* \in \mathbb{C}_{\geq}^{n},$$

which imply $G^{\#} = G$ and $H^{\#} = H$ according to (1.3). ∎

The decomposition (3.3) is called the (WPD) of the rectangular complex matrix A, and E and G, H are called the and the of this decomposition, respectively. From the proof of Theorem 3.5, we find that WPD may not be unique. The uniqueness of WPD can be obtained through the right WPD (3.6) and the left WPD (3.11) as follows.

THEOREM 3.6. $A \in \mathbb{C}_r^{m\times n}$ A

$$A = GE \quad R(A) \subseteq R(E), \tag{3.6}$$

$E \in \mathbb{C}^{m\times n}$ $MG \in \mathbb{C}_{\geq}^{m}$ G $G^{\#} = G$ G E $r(E) = r(A)$ $E = ((AA^{\#})^{1/2})_{MM}^{+} A = A((A^{\#}A)^{1/2})_{NN}^{+}$ $r(G) = r(A)$ G $G = (AA^{\#})^{1/2}$ G E (3.6) $A = (AA^{\#})^{1/2}E$

. From Theorem 3.5, we need only to prove the second part of this theorem.

Since E is an MN-WPI matrix and $R(A) \subseteq R(E)$, from Lemma 3.3, we have

$$A = EE^{\#}A = EE^{\#}GE. \tag{3.7}$$

Then we can get

$$\begin{aligned}(AA^{\#})^{1/2} &= (EE^{\#}GEE^{\#}GEE^{\#})^{1/2} = EE^{\#}GEE^{\#}\\ &\Leftrightarrow\ (AA^{\#})^{1/2}E = AE^{\#}E, \qquad (3.8)\\ (A^{\#}A)^{1/2} &= (E^{\#}GEE^{\#}EE^{\#}GE)^{1/2} = E^{\#}GE\\ &\Leftrightarrow\ E(A^{\#}A)^{1/2} = A. \qquad (3.9)\end{aligned}$$

Let $r(E) = r(A)$. Then $R(A) = R(E)$ and $r(E^{\#}) = r(A^{\#})$. Meanwhile, observing that $R(A^{\#}) \subseteq R(E^{\#})$ from $A = GE$, we have $R(A^{\#}) = R(E^{\#})$. Thus, according to (3.8) and the fact that $R((AA^{\#})^{1/2}) = R(A) = R(E)$, we can get

$$\begin{aligned}&(AA^{\#})^{1/2}E = AE^{\#}E = A\\ &\Leftrightarrow ((AA^{\#})^{1/2})^{+}_{MM}(AA^{\#})^{1/2}E = ((AA^{\#})^{1/2})^{+}_{MM}A\\ &\Leftrightarrow (AA^{\#})^{1/2}((AA^{\#})^{1/2})^{+}_{MM}E = ((AA^{\#})^{1/2})^{+}_{MM}A\\ &\Leftrightarrow E = ((AA^{\#})^{1/2})^{+}_{MM}A.\end{aligned}$$

Similarly, from (3.9) and the fact that $R((A^{\#}A)^{1/2}) = R(A^{\#}) = R(E^{\#})$, we can get

$$E = A((A^{\#}A)^{1/2})^{+}_{NN}.$$

Therefore, E is uniquely determined and

$$E = ((AA^{\#})^{1/2})^{+}_{MM}A = A((A^{\#}A)^{1/2})^{+}_{NN}.$$

Let $r(G) = r(A)$. Then, in view of the fact that $A = GE$, we have $R(G) = R(A)$, which together with $R(A) \subseteq R(E)$ gives

$$R(G) \subseteq R(E). \tag{3.10}$$

Thus, by (3.8) and (3.10), we conclude that

$$(AA^{\#})^{1/2} = AE^{\#} \ \Leftrightarrow\ (AA^{\#})^{1/2} = GEE^{\#} \ \Leftrightarrow\ (AA^{\#})^{1/2} = EE^{\#}G = G.$$

Therefore, G is uniquely determined and $G = (AA^{\#})^{1/2}$. Moreover, given G and E in (3.6), $A = (AA^{\#})^{1/2}E$ can be obtained by (3.7) and (3.8). □

Similar to the right WPD, we can present the left WPD as follows.

THEOREM 3.7. [illegible] $A \in \mathbb{C}_r^{m\times n}$ [illegible] A [illegible]

$$A = EH \text{ [illegible] } R(A^{\#}) \subseteq R(E^{\#}), \tag{3.11}$$

[illegible] $E \in \mathbb{C}^{m\times n}$ [illegible] $NH \in \mathbb{C}^{n}_{\geq}$ [illegible] H [illegible]
[illegible] $H^{\#} = H$
[illegible] H [illegible] E [illegible]

$r(E) = r(A)$ E $E = A((A^{\#}A)^{1/2})^{+}_{NN} = ((AA^{\#})^{1/2})^{+}_{MM}A$

$r(H) = r(A)$ H $H = (A^{\#}A)^{1/2}$

E H (3.11) $A = E(A^{\#}A)^{1/2}$

The following unique WPD theorem follows from Theorems 3.6 and 3.7.

THEOREM 3.8. $A \in \mathbb{C}_r^{m\times n}$ $G = (AA^{\#})^{1/2}$ $H = (A^{\#}A)^{1/2}$

$$A = GE = EH, \tag{3.12}$$

$E = ((AA^{\#})^{1/2})^{+}_{MM}A = A((A^{\#}A)^{1/2})^{+}_{NN}$

Considering the WPDs of two rectangular complex matrices, we introduce the simultaneous weighted polar decomposition (SWPD) as follows.

DEFINITION 3.9. $A, B \in \mathbb{C}^{m\times n}$ $E \in \mathbb{C}^{m\times n}$ $A = G_1E = EH_1$ $B = G_2E = EH_2$ $G_1, G_2 \in \mathbb{C}^{m\times m}$ $H_1, H_2 \in \mathbb{C}^{n\times n}$

From the proof of Theorem 3.5 and the simultaneous (M, N) weighted singular value decomposition (MN-SSVD) [16], we can obtain the following sufficient condition for SWPD.

THEOREM 3.10. $A, B \in \mathbb{C}^{m\times n}$ A, B (M, N) A, B

The following lemma provides a necessary and sufficient condition for MN-SSVD given by Liu [16]. From this lemma, another sufficient condition for SWPD can be presented.

LEMMA 3.11. $A, B \in \mathbb{C}^{m\times n}$ A, B (M, N) $(A^{\#}B)^{\#} = A^{\#}B$, $(AB^{\#})^{\#} = AB^{\#}$ $NA^{\#}B \in \mathbb{C}^n_{\geq}$, $MAB^{\#} \in \mathbb{C}^m_{\geq}$

COROLLARY 3.12. $A, B \in \mathbb{C}^{m\times n}$ $(A^{\#}B)^{\#} = A^{\#}B$, $(AB^{\#})^{\#} = AB^{\#}$ $NA^{\#}B \in \mathbb{C}^n_{\geq}$, $MAB^{\#} \in \mathbb{C}^m_{\geq}$ A, B

In order to derive further criteria for SWPD, next we will show that SWPD can be equivalently defined via left SWPD and right SWPD, respectively.

DEFINITION 3.13. $A, B \in \mathbb{C}^{m\times n}$

(a) A, B $E \in \mathbb{C}^{m\times n}$ $A = EH_1$ $R(A^{\#}) \subseteq R(E^{\#})$ $B = EH_2$ $R(B^{\#}) \subseteq R(E^{\#})$ $H_1, H_2 \in \mathbb{C}^{n\times n}$

(b) A, B $E \in \mathbb{C}^{m\times n}$ $A = G_1E$ $R(A) \subseteq R(E)$ $B = G_2E$ $R(B) \subseteq R(E)$ $G_1, G_2 \in \mathbb{C}^{m\times m}$

LEMMA 3.14. $A, B \in \mathbb{C}^{m\times n}$ A, B

. Let A, B be simultaneously left weighted polar decomposable. From Definition 3.13(a), we have $R(A) \subseteq R(E)$ and $R(B) \subseteq R(E)$. Furthermore, observe that E is an MN-WPI matrix and $R(A^{\#}) \subseteq R(E^{\#})$, $R(B^{\#}) \subseteq R(E^{\#})$. Then we have

$$A = EH_1 \Rightarrow A^{\#} = H_1E^{\#} \Rightarrow E^{\#}EA^{\#} = E^{\#}EH_1E^{\#}$$

$$\Rightarrow A^{\#} = E^{\#}EH_1E^{\#} \Rightarrow A = G_1E,$$

$$B = EH_2 \Rightarrow B^{\#} = H_2E^{\#} \Rightarrow E^{\#}EB^{\#} = E^{\#}EH_2E^{\#}$$

$$\Rightarrow B^{\#} = E^{\#}EH_2E^{\#} \Rightarrow B = G_2E,$$

where $G_1 = EH_1E^{\#}$ and $G_2 = EH_2E^{\#}$ are generalized positive semidefinite matrices. Thus, we get that A, B are simultaneously right weighted polar decomposable. Now the proof can be completed using similar arguments. □

From Theorem 3.6 and Lemma 3.14, we can easily deduce the following theorem.

THEOREM 3.15. [illegible] $A, B \in \mathbb{C}^{m\times n}$ [illegible] A, B [illegible] $E \in \mathbb{C}^{m\times n}$ [illegible]

$$A = (AA^{\#})^{1/2}E, R(A) \subseteq R(E) \text{ and } B = (BB^{\#})^{1/2}E, R(B) \subseteq R(E).$$

Another characterization (Theorem 4.9) for SWPD is associated with a new matrix partial ordering which will be discussed in the following section.

4. WGL partial ordering. Before introducing WGL partial ordering, we first define the following partial ordering of complex matrices which is a generalization of the Löwner partial ordering.

DEFINITION 4.1. [illegible] $A, B \in \mathbb{C}^{m\times m}$ [illegible] A [illegible] B [illegible] $A \leq_{WL} B$ [illegible] $M(B - A) \in \mathbb{C}^m_{\geq}$

The WL partial ordering, important for studying the WGL partial ordering, can be interpreted as the weighted Löwner partial ordering.

A similar definition and some characteristics can be found in [7]. Those characteristics can be used to prove some of our results. However, it will increase the complexity of proof to some extent; therefore, in this paper we use alternative methods to obtain these results.

Now we present the definition of WGL partial ordering formally.

DEFINITION 4.2. [illegible] $A, B \in \mathbb{C}^{m\times n}$ [illegible] A [illegible] B [illegible] WGL [illegible] $A \leq_{WGL} B$ [illegible] $(AA^{\#})^{1/2} \leq_{WL} (BB^{\#})^{1/2}$ [illegible] $AB^{\#} = (AA^{\#})^{1/2}(BB^{\#})^{1/2}$

Next we show that the relation $\leq_{WGL}$ satisfies the three laws of the matrix partial ordering.

(1) $A \leq_{WGL} A$ holds obviously.

(2) If $A \leq_{WGL} B$ and $B \leq_{WGL} C$, to verify $A \leq_{WGL} C$, in view of Definition 4.1, we need only to verify that $AC^{\#} = (AA^{\#})^{1/2}(CC^{\#})^{1/2}$.

From Definition 4.2, Definition 4.1, and Lemma 2.2, we have

$$M(AA^{\#})^{1/2} = M(BB^{\#})^{1/2}K^{\#} \tag{4.1}$$

$$\Leftrightarrow (AA^{\#})^{1/2}M = K(BB^{\#})^{1/2}M \quad \text{for some } K \in \mathbb{C}^{m\times m}.$$

Meanwhile, note that $R(A^{\#}) \subseteq R(B^{\#})$ (see Theorem 4.5 and Lemma 4.3). Hence,

$$A^{\#} = B^{+}_{MN}BA^{\#} \Rightarrow A = AB^{+}_{MN}B. \tag{4.2}$$

Thus, according to (4.1) and (4.2), we conclude that

$$\begin{aligned}
AC^{\#} &= AB^{+}_{MN}BB^{+}_{MN}BC^{\#} = AB^{\#}(B^{+}_{MN})^{\#}B^{+}_{MN}BC^{\#} \\
&= (AA^{\#})^{1/2}(BB^{\#})^{1/2}(B^{+}_{MN})^{\#}B^{+}_{MN}BC^{\#} \\
&= (AA^{\#})^{1/2}MM^{-1}(BB^{\#})^{1/2}(B^{+}_{MN})^{\#}B^{+}_{MN}(BB^{\#})^{1/2}(CC^{\#})^{1/2} \\
&= K(BB^{\#})^{1/2}MM^{-1}(BB^{\#})^{1/2}(B^{+}_{MN})^{\#}B^{+}_{MN}(BB^{\#})^{1/2}(CC^{\#})^{1/2} \\
&= KB(B^{+}_{MN}B)^{\#}B^{+}_{MN}(BB^{\#})^{1/2}(CC^{\#})^{1/2} \\
&= KBB^{+}_{MN}(BB^{\#})^{1/2}(CC^{\#})^{1/2} \\
&= K(BB^{\#})^{1/2}MM^{-1}(CC^{\#})^{1/2} \\
&= (AA^{\#})^{1/2}(CC^{\#})^{1/2}.
\end{aligned}$$

(3) If $A \leq_{WGL} B$, $B \leq_{WGL} A$, then

$$(A-B)(A-B)^{\#} = AA^{\#} - AB^{\#} - BA^{\#} + BB^{\#} = 0,$$

which implies $A = B$.

Therefore, the relation $\leq_{WGL}$ is a partial ordering of matrices. From Definition 4.1, the WGL partial ordering can be viewed as the weighted GL partial ordering [8].

The following lemma is useful for studying the properties of WGL partial ordering later in this paper.

LEMMA 4.3. [illegible] $A, B \in \mathbb{C}^{m\times n}$ [illegible]

[illegible] $AB^{\#} = (AA^{\#})^{1/2}(BB^{\#})^{1/2}$ [illegible] $R(A) \subseteq R(B)$, [illegible] $R(A^{\#}) \subseteq R(B^{\#})$.

[illegible]. Premultiplying the equality $AB^{\#} = (AA^{\#})^{1/2}(BB^{\#})^{1/2}$ by A^{+}_{MN} and postmultiplying it by $(B^{+}_{MN})^{\#}A^{\#}(A^{+}_{MN})^{\#}$ gives

$$A^{+}_{MN}AB^{\#}(B^{+}_{MN})^{\#}A^{\#}(A^{+}_{MN})^{\#} = A^{+}_{MN}(AA^{\#})^{1/2}(BB^{\#})^{1/2}(B^{+}_{MN})^{\#}A^{\#}(A^{+}_{MN})^{\#},$$

which is equivalent to

$$P_{R(A^{\#})}P_{R(B^{\#})}P_{R(A^{\#})} = A^{+}_{MN}(AA^{\#})^{1/2}(BB^{\#})^{1/2}(B^{+}_{MN})^{\#}A^{\#}(A^{+}_{MN})^{\#}, \tag{4.3}$$

while from $R(A) \subseteq R(B)$ and the fact that $AB^{\#} = (AA^{\#})^{1/2}(BB^{\#})^{1/2}$, we have

$$\begin{aligned}
(BB^{\#})^{1/2}(B^{+}_{MN})^{\#}A^{\#} &= (BB^{\#})^{1/2}(AB^{+}_{MN})^{\#} = (BB^{\#})^{1/2}(AB^{+}_{MN}BB^{+}_{MN})^{\#} \\
&= (BB^{\#})^{1/2}(B^{+}_{MN})^{\#}(B^{+}_{MN}B)^{\#}A^{\#} = (BB^{\#})^{1/2}(B^{+}_{MN})^{\#}B^{+}_{MN}BA^{\#} \\
&= (BB^{\#})^{1/2}(B^{+}_{MN})^{\#}B^{+}_{MN}(BB^{\#})^{1/2}(AA^{\#})^{1/2} \\
&= (BB^{\#})^{1/2}(BB^{\#})^{+}_{MN}(BB^{\#})^{1/2}(AA^{\#})^{1/2} \\
&= (AA^{\#})^{1/2},
\end{aligned}$$

which combined with (4.3) leads to

$$\begin{aligned}
P_{R(A^{\#})}P_{R(B^{\#})}P_{R(A^{\#})} &= A^{+}_{MN}(AA^{\#})^{1/2}(AA^{\#})^{1/2}(A^{+}_{MN})^{\#} \\
&= A^{+}_{MN}AA^{\#}(A^{+}_{MN})^{\#} = P_{R(A^{\#})}.
\end{aligned}$$

Therefore,

$$0 = P_{R(A^{\#})}P_{R(B^{\#})}P_{R(A^{\#})} - P_{R(A^{\#})} = P_{R(A^{\#})}(I - P_{R(B^{\#})})P_{R(A^{\#})}$$

$$\Leftrightarrow (I - P_{R(B^{\#})})P_{R(A^{\#})} = 0$$

$$\Leftrightarrow R(P_{R(A^{\#})}) \subseteq N(I - P_{R(B^{\#})}) \Leftrightarrow R(A^{\#}) \subseteq N(I - P_{R(B^{\#})}). \tag{4.4}$$

Observe that the matrix $I - P_{R(B^{\#})}$ is idempotent. Hence, $N(I - P_{R(B^{\#})}) = R(P_{R(B^{\#})}) = R(B^{\#})$, which together with (4.4) gives $R(A^{\#}) \subseteq R(B^{\#})$. □

In the following, we discuss the properties of WGL partial ordering.

THEOREM 4.4. [illegible] $A, B \in \mathbb{C}^{m\times n}$ [illegible]

$$A \leq_{WGL} B \text{ [illegible] } A^{\#} \leq_{WGL} B^{\#}.$$

[illegible]. Let $r(A) = a$ and $r(B) = b$. From Lemma 2.1, we suppose that the MN-SVDs for A and B are $A = U_a\Sigma_a V_a^*$ and $B = U_b\Sigma_b V_b^*$, where $\Sigma_a \in \mathbb{C}_>^a$ and $\Sigma_b \in \mathbb{C}_>^b$ are diagonal matrices, and $U_a \in \mathbb{C}^{m\times a}$, $U_b \in \mathbb{C}^{m\times b}$, $V_a \in \mathbb{C}^{n\times a}$, $V_b \in \mathbb{C}^{n\times b}$ satisfy $U_a^*MU_a = V_a^*N^{-1}V_a = I_a$ and $U_b^*MU_b = V_b^*N^{-1}V_b = I_b$.

From Definition 4.1 and Definition 4.2, we can get $A \leq_{WGL} B$ if and only if

(a) $(AA^{\#})^{1/2} \leq_{WL} (BB^{\#})^{1/2}$

$$\Leftrightarrow M(U_a\Sigma_a V_a^*N^{-1}V_a\Sigma_a U_a^*M)^{1/2} \leq_L M(U_b\Sigma_b V_b^*N^{-1}V_b\Sigma_b U_b^*M)^{1/2}$$

$$\Leftrightarrow M(U_a\Sigma_a U_a^*MU_a\Sigma_a U_a^*M)^{1/2} \leq_L M(U_b\Sigma_b U_b^*MU_b\Sigma_b U_b^*M)^{1/2}$$

$$\Leftrightarrow MU_a\Sigma_a U_a^*M \leq_L MU_b\Sigma_b U_b^*M \Leftrightarrow U_a\Sigma_a U_a^* \leq_L U_b\Sigma_b U_b^*, \tag{4.5}$$

(b) $AB^{\#} = (AA^{\#})^{1/2}(BB^{\#})^{1/2}$

$$\Leftrightarrow U_a\Sigma_a V_a^*N^{-1}V_b\Sigma_b U_b^*M = U_a\Sigma_a U_a^*MU_b\Sigma_b U_b^*M$$

$$\Leftrightarrow V_a^*N^{-1}V_b = U_a^*MU_b, \tag{4.6}$$

and $A^{\#} \leq_{WGL} B^{\#}$ if and only if

(c) $(A^{\#}A)^{1/2} \leq_{WL} (B^{\#}B)^{1/2}$

$$\Leftrightarrow N(N^{-1}V_a\Sigma_a U_a^*MU_a\Sigma_a V_a^*)^{1/2} \leq_L N(N^{-1}V_b\Sigma_b U_b^*MU_b\Sigma_b V_b^*)^{1/2}$$

$$\Leftrightarrow N(N^{-1}V_a\Sigma_a V_a^*N^{-1}V_a\Sigma_a V_a^*)^{1/2} \leq_L N(N^{-1}V_b\Sigma_b V_b^*N^{-1}V_b\Sigma_b V_b^*)^{1/2}$$

$$\Leftrightarrow NN^{-1}V_a\Sigma_a V_a^* \leq_L NN^{-1}V_b\Sigma_b V_b^* \Leftrightarrow V_a\Sigma_a V_a^* \leq_L V_b\Sigma_b V_b^*, \tag{4.7}$$

(d) $A^{\#}B = (A^{\#}A)^{1/2}(B^{\#}B)^{1/2}$

$$\Leftrightarrow N^{-1}V_a\Sigma_a U_a^*MU_b\Sigma_b V_b^* = N^{-1}V_a\Sigma_a V_a^*N^{-1}V_b\Sigma_b V_b^*$$

$$\Leftrightarrow U_a^*MU_b = V_a^*N^{-1}V_b. \tag{4.8}$$

Then, by (4.5), (4.6), (4.7), and (4.8), we have $A \leq_{WGL} B \Leftrightarrow A^{\#} \leq_{WGL} B^{\#}$ if and only if the following relation (4.9) holds under the condition (4.6) (or (4.8)):

$$U_a\Sigma_a U_a^* \leq_L U_b\Sigma_b U_b^* \Leftrightarrow V_a\Sigma_a V_a^* \leq_L V_b\Sigma_b V_b^*. \tag{4.9}$$

According to $A \leq_{WGL} B$, we can get $R(A^{\#}) \subseteq R(B^{\#})$ (see Theorem 4.5 and Lemma 4.3), which combined with the MN-SVDs of A and B gives

$$R(N^{-1}V_a\Sigma_a U_a^* M) \subseteq R(N^{-1}V_b\Sigma_b U_b^* M). \tag{4.10}$$

Note that

$$r(N^{-1}V_i\Sigma_i U_i^* M) = r(N^{-1}V_i), \quad R(N^{-1}V_i\Sigma_i U_i^* M) \subseteq R(N^{-1}V_i), \qquad i = a, b,$$

and hence, $R(N^{-1}V_i D_i U_i^* M) = R(N^{-1}V_i)$, $i = a, b$. Therefore, $R(N^{-1}V_a) \subseteq R(N^{-1}V_b)$. Thus there exists a matrix H such that $N^{-1}V_a = N^{-1}V_b H$. Premultiplying the last equality by $N^{1/2}$, we can get

$$N^{-1/2}V_a = N^{-1/2}V_b H.$$

Consequently,

$$R(N^{-1/2}V_a) \subseteq R(N^{-1/2}V_b). \tag{4.11}$$

Note that $N^{-1/2}V_b(N^{-1/2}V_b)^* N^{-1/2}V_b = N^{-1/2}V_b$. Then $N^{-1/2}V_b$ is a partial isometric matrix [2]. Thus, by (4.11), we have

$$N^{-1/2}V_b(N^{-1/2}V_b)^* N^{-1/2}V_a = N^{-1/2}V_a, \text{ i.e., } V_b V_b^* N^{-1}V_a = V_a. \tag{4.12}$$

According to Lemma 2.3, and considering (4.6) and (4.12), we can conclude that

$$\begin{aligned}
U_a\Sigma_a U_a^* \leq_L U_b\Sigma_b U_b^* &\Rightarrow U_b^* M U_a\Sigma_a U_a^* M U_b \leq_L U_b^* M U_b\Sigma_b U_b^* M U_b \\
&\Rightarrow U_b^* M U_a\Sigma_a U_a^* M U_b \leq_L \Sigma_b \\
&\Rightarrow V_b U_b^* M U_a\Sigma_a U_a^* M U_b V_b^* \leq_L V_b\Sigma_b V_b^* \\
&\Rightarrow V_b V_b^* N^{-1}V_a\Sigma_a V_a^* N^{-1}V_b V_b^* \leq_L V_b\Sigma_b V_b^* \\
&\Rightarrow V_a\Sigma_a V_a^* \leq_L V_b\Sigma_b V_b^*.
\end{aligned}$$

In the similar way, we can get $U_b U_b^* M U_a = U_a$ through $A^{\#} \leq_{WGL} B^{\#}$. Thus from Lemma 2.3 and (4.6), we have

$$\begin{aligned}
V_a\Sigma_a V_a^* \leq_L V_b\Sigma_b V_b^* &\Rightarrow V_b V_b^* N^{-1}V_a\Sigma_a V_a^* N^{-1}V_b V_b^* \leq_L V_b\Sigma_b V_b^* \\
&\Rightarrow V_b U_b^* M U_a\Sigma_a U_a^* M U_b V_b^* \leq_L V_b\Sigma_b V_b^* \\
&\Rightarrow U_b^* M U_a\Sigma_a U_a^* M U_b \leq_L \Sigma_b \\
&\Rightarrow U_b U_b^* M U_a\Sigma_a U_a^* M U_b U_b^* \leq_L U_b\Sigma_b U_b^* \\
&\Rightarrow U_a\Sigma_a U_a^* \leq_L U_b\Sigma_b U_b^*. \qquad \square
\end{aligned}$$

THEOREM 4.5. [illegible] $A, B \in \mathbb{C}^{m\times n}$ [illegible]

$$A \leq_{WGL} B$$

[illegible]

$$\lambda_1(B_{MN}^+ A) \leq 1, \ R(A) \subseteq R(B), \text{ [illegible] } AB^{\#} = (AA^{\#})^{1/2}(BB^{\#})^{1/2}.$$

. According to Lemma 2.2 and the proof of Theorem 4.4 (using the symbols in Theorem 4.4), we can get

$$(AA^{\#})^{1/2} \leq_{WL} (BB^{\#})^{1/2} \Leftrightarrow M^{1/2}U_a\Sigma_a U_a^* M^{1/2} \leq_L M^{1/2}U_b\Sigma_b U_b^* M^{1/2}$$

$$\Leftrightarrow \begin{cases} \lambda_1(((M^{1/2}U_b)\Sigma_b(M^{1/2}U_b)^*)^+ M^{1/2}U_a\Sigma_a U_a^* M^{1/2}) \leq 1, \\ R(M^{1/2}U_a\Sigma_a U_a^* M^{1/2}) \subseteq R(M^{1/2}U_b\Sigma_b U_b^* M^{1/2}). \end{cases} \tag{4.13}$$

Observing (4.6), i.e., $V_a^* N^{-1} V_b = U_a^* M U_b$, and Lemma 2.4, we have

$$\begin{aligned} &\lambda_1(((M^{1/2}U_b)\Sigma_b(M^{1/2}U_b)^*)^+ M^{1/2}U_a\Sigma_a U_a^* M^{1/2}) \\ &= \lambda_1(M^{1/2}U_b\Sigma_b^{-1}U_b^* M U_a\Sigma_a U_a^* M^{1/2}) = \lambda_1(U_a^* M U_b\Sigma_b^{-1}U_b^* M U_a\Sigma_a) \\ &= \lambda_1(V_a^* N^{-1} V_b\Sigma_b^{-1}U_b^* M U_a\Sigma_a) = \lambda_1(N^{-1}V_b\Sigma_b^{-1}U_b^* M U_a\Sigma_a V_a^*) \\ &= \lambda_1(B_{MN}^+ A) \leq 1. \end{aligned} \tag{4.14}$$

Meanwhile, by (4.13), we can also get that there exists a matrix $H \in \mathbb{C}^{m\times m}$ such that

$$M^{1/2}U_a\Sigma_a U_a^* M^{1/2} = M^{1/2}U_b\Sigma_b U_b^* M^{1/2} H.$$

Therefore,

$$U_a\Sigma_a U_a^* M^{1/2} = U_b\Sigma_b U_b^* M^{1/2} H \text{ and } R(U_a\Sigma_a U_a^* M^{1/2}) \subseteq R(U_b\Sigma_b U_b^* M^{1/2}).$$

As a result,

$$R(U_a) \subseteq R(U_b),$$

which together with the MN-SVDs of A and B gives $R(A) \subseteq R(B)$. Then the proof is completed in view of Definition 4.2. □

The weighted star partial ordering of matrices was characterized by Liu using MN-SSVD in [16]. Its characteristics can be useful in comparing the WGL partial ordering with the weighted star partial ordering. The following result is from [16].

LEMMA 4.6. . $A, B \in \mathbb{C}^{m\times n}$. $r(B) = b > r(A) = a \geq 1$. $A \overset{\#}{\leq} B$ . . $U \in \mathbb{C}^{m\times m}$. $V \in \mathbb{C}^{n\times n}$. $U^*MU = I_m$. $V^*N^{-1}V = I_n$.

$$A = U\begin{pmatrix} \Sigma_a & 0 \\ 0 & 0 \end{pmatrix} V^*, \quad B = U\begin{pmatrix} \Sigma_a & 0 & 0 \\ 0 & \Sigma & 0 \\ 0 & 0 & 0 \end{pmatrix} V^*, \tag{4.15}$$

. $\Sigma_a \in \mathbb{C}_>^a$. $\Sigma \in \mathbb{C}_>^{b-a}$

The following theorem is a straightforward consequence of Lemma 4.6, Definition 4.1, and Definition 4.2.

THEOREM 4.7. . $A, B \in \mathbb{C}^{m\times n}$ . .

. $A \overset{\#}{\leq} B$, . . $A \leq_{WGL} B$.

In the next two theorems, we will study WGL partial ordering by WPD and discuss the characteristic for SWPD via WGL partial ordering.

THEOREM 4.8. Let $A, B \in \mathbb{C}^{m\times n}$ be such that $A = G_1E_1, B = G_2E_2$ are the weighted polar decompositions of A and B, respectively, where $G_1 = (AA^{\#})^{1/2}$ and $G_2 = (BB^{\#})^{1/2}$. Then

$$A \leq_{WGL} B \text{ if and only if } G_1 \leq_{WL} G_2 \text{ and } E_1 \overset{\#}{\leq} E_2.$$

Proof. From Definition 4.2, we have

$$A \leq_{WGL} B \Leftrightarrow \begin{cases} (AA^{\#})^{1/2} \leq_{WL} (BB^{\#})^{1/2}, \\ AB^{\#} = (AA^{\#})^{1/2}(BB^{\#})^{1/2} \end{cases} \Leftrightarrow \begin{cases} G_1 \leq_{WL} G_2, \\ AB^{\#} = (AA^{\#})^{1/2}(BB^{\#})^{1/2}. \end{cases}$$

However,

$$AB^{\#} = (AA^{\#})^{1/2}(BB^{\#})^{1/2} \Leftrightarrow G_1E_1E_2^{\#}G_2 = G_1G_2. \tag{4.16}$$

Premultiplying the second equality of (4.16) by $G_{1,MM}^{+}$ and postmultiplying it by $G_{2,MM}^{+}$, and observing the following equalities:

$$\begin{cases} G_1G_{1,MM}^{+} = G_{1,MM}^{+}G_1 = E_1E_1^{\#} = P_{R(A)}, \\ G_2G_{2,MM}^{+} = G_{2,MM}^{+}G_2 = E_2E_2^{\#} = P_{R(B)}, \end{cases} \tag{4.17}$$

we can get

$$G_{1,MM}^{+}G_1E_1E_2^{\#}G_2G_{2,MM}^{+} = G_{1,MM}^{+}G_1G_2G_{2,MM}^{+} \Leftrightarrow E_1E_2^{\#} = E_1E_1^{\#}E_2E_2^{\#}. \tag{4.18}$$

According to the fact that $G_1 \leq_{WL} G_2$ and the proof of Theorem 4.5, we have

$$R(A) \subseteq R(B),$$

which combined with (4.17) and (4.18) gives

$$P_{R(B)}P_{R(A)} = P_{R(A)} \Leftrightarrow E_2E_2^{\#}E_1E_1^{\#} = E_1E_1^{\#} \Leftrightarrow E_2E_1^{\#} = E_1E_1^{\#}. \tag{4.19}$$

Similarly, according to the fact that $G_1 \leq_{WL} G_2$, the proof of Theorem 4.5, the first equality of (4.16), and Lemma 4.3, we can get

$$R(A^{\#}) \subseteq R(B^{\#}).$$

Observing that $E_1^{\#}E_1 = P_{R(A^{\#})}$, $E_2^{\#}E_2 = P_{R(B^{\#})}$, and (4.19), we have

$$P_{R(B^{\#})}P_{R(A\#)} = P_{R(A\#)} \Leftrightarrow E_2^{\#}E_2E_1^{\#}E_1 = E_1^{\#}E_1 \Leftrightarrow E_2^{\#}E_1 = E_1^{\#}E_1.$$

Therefore, $E_1^{\#}E_2 = E_1^{\#}E_1$, which combined with (4.19) leads to $E_1 \overset{\#}{\leq} E_2$.

Conversely, according to $E_1 \overset{\#}{\leq} E_2$, we have $E_2E_1^{\#} = E_1E_1^{\#}$, which together with (4.17) gives $G_1E_1E_2^{\#}G_2 = G_1G_2$. Thus, the proof is completed. ∎

THEOREM 4.9. Let $A, B \in \mathbb{C}^{m\times n}$ be such that $AB^{\#} = (AA^{\#})^{1/2}(BB^{\#})^{1/2}$.

Proof. The necessity of the condition follows from Theorem 3.15. Now we prove the sufficiency.

Since $AB^{\#} = (AA^{\#})^{1/2}(BB^{\#})^{1/2}$, we have

$$BA^{\#} = (AB^{\#})^{\#} = ((AA^{\#})^{1/2}(BB^{\#})^{1/2})^{\#} = (BB^{\#})^{1/2}(AA^{\#})^{1/2}.$$

As a result,

$$((A+B)(A+B)^{\#})^{1/2} = (AA^{\#} + AB^{\#} + BA^{\#} + BB^{\#})^{1/2}$$

$$= (AA^{\#})^{1/2} + (BB^{\#})^{1/2}. \tag{4.20}$$

Then the following relations (4.21) and (4.22) hold:

$$A(A+B)^{\#} = (AA^{\#})^{1/2}((A+B)(A+B)^{\#})^{1/2}, \tag{4.21}$$

$$(BB^{\#})^{1/2} = ((A+B)(A+B)^{\#})^{1/2} - (AA^{\#})^{1/2}, \tag{4.22}$$

where the equality (4.22) implies

$$M(BB^{\#})^{1/2} = M((A+B)(A+B)^{\#})^{1/2} - M(AA^{\#})^{1/2}. \tag{4.23}$$

According to the proof of Theorem 4.4 (using the symbols of Theorem 4.4), we know that $M(BB^{\#})^{1/2} = MU_b\Sigma_b U_b^* M \in \mathbb{C}^m_{\geq}$, which together with (4.23) and Definition 4.1 leads to

$$(AA^{\#})^{1/2} \leq_{WL} ((A+B)(A+B)^{\#})^{1/2}. \tag{4.24}$$

Similarly, we can get that

$$B(A+B)^{\#} = (BB^{\#})^{1/2}((A+B)(A+B)^{\#})^{1/2}, \tag{4.25}$$

$$(BB^{\#})^{1/2} \leq_{WL} ((A+B)(A+B)^{\#})^{1/2}. \tag{4.26}$$

Then, according to (4.21), (4.24), (4.25), (4.26), and Definition 4.2, we conclude that

$$A \leq_{WGL} (A+B) \text{ and } B \leq_{WGL} (A+B),$$

and hence from Theorem 4.8, we have

$$E_1 \overset{\#}{\leq} E \text{ and } E_2 \overset{\#}{\leq} E, \tag{4.27}$$

where

$$E_1 = ((AA^{\#})^{1/2})^{+}_{MM} A = A((A^{\#}A)^{1/2})^{+}_{NN},$$

$$E_2 = ((BB^{\#})^{1/2})^{+}_{MM} B = B((B^{\#}B)^{1/2})^{+}_{NN},$$

$$E = (((A+B)(A+B)^{\#})^{1/2})^{+}_{MM}(A+B)$$

$$= (A+B)(((A+B)^{\#}(A+B))^{1/2})^{+}_{NN}.$$

According to the fact that $A = (AA^{\#})^{1/2}((AA^{\#})^{1/2})^{+}_{MM} A$ and (4.27), we can obtain that

$$AE^{\#} = (AA^{\#})^{1/2}E_1E_1^{\#} = (AA^{\#})^{1/2}((AA^{\#})^{1/2})^{+}_{MM} AA^{\#}((AA^{\#})^{1/2})^{+}_{MM}$$

$$= (AA^{\#})^{1/2}(AA^{\#})^{1/2}((AA^{\#})^{1/2})^{+}_{MM} = (AA^{\#})^{1/2}.$$

Therefore,

$$AE^{\#}E = (AA^{\#})^{1/2}E. \tag{4.28}$$

Meanwhile, according to the fact that

$$A = G_1E_1 \;\Leftrightarrow\; A^{\#} = E_1^{\#}G_1 \;\Rightarrow\; R(A^{\#}) \subseteq R(E_1^{\#})$$

and (4.27), we can get $R(A^{\#}) \subseteq R(E^{\#})$, which combined with (4.28) gives

$$A = (AA^{\#})^{1/2}E. \tag{4.29}$$

Similarly, according to $E_1 = A((A^{\#}A)^{1/2})^{+}_{NN}$, we can derive that

$$EE^{\#}A = E(A^{\#}A)^{1/2} \;\Leftrightarrow\; A = E(A^{\#}A)^{1/2},$$

and according to $E_2 = ((BB^{\#})^{1/2})^{+}_{MM}B = B((B^{\#}B)^{1/2})^{+}_{NN}$, we can derive that

$$BE^{\#} = (BB^{\#})^{1/2} \;\Leftrightarrow\; BE^{\#}E = (BB^{\#})^{1/2}E \;\Leftrightarrow\; B = (BB^{\#})^{1/2}E, \tag{4.30}$$

$$E^{\#}B = (B^{\#}B)^{1/2} \;\Leftrightarrow\; EE^{\#}B = E(B^{\#}B)^{1/2} \;\Leftrightarrow\; B = E(B^{\#}B)^{1/2}. \tag{4.31}$$

Further, since $AE^{\#} = EA^{\#} = (AA^{\#})^{1/2}$, we have $R((AA^{\#})^{1/2}) \subseteq R(E)$. Thus, noting the fact that $R((AA^{\#})^{1/2}) = R(A)$, we can get

$$R(A) \subseteq R(E). \tag{4.32}$$

Similarly, we can get

$$R(B) \subseteq R(E). \tag{4.33}$$

In view of (4.29), (4.30), (4.32), (4.33), and Theorem 3.15, we can see that the sufficiency of the condition holds. □

5. Methods for computing the WPD. From Lemma 2.1 and Theorem 3.8, the generalized positive semidefinite polar factor G can be computed by

$$\begin{aligned} G &= (AA^{\#})^{1/2} = (U_r\Sigma V_r^* N^{-1} V_r \Sigma U_r^* M)^{1/2} \\ &= (U_r\Sigma U_r^* M U_r \Sigma U_r^* M)^{1/2} = U_r\Sigma U_r^* M. \end{aligned} \tag{5.1}$$

Similarly, H can be computed by

$$H = (A^{\#}A)^{1/2} = N^{-1}V_r\Sigma V_r^*. \tag{5.2}$$

Then, we have

$$\begin{aligned} E &= ((AA^{\#})^{1/2})^{+}_{MM}A = A((A^{\#}A)^{1/2})^{+}_{NN} \\ &= (U_r\Sigma U_r^* M)^{+}_{MM}U_r\Sigma V_r^* = U_r\Sigma V_r^*(N^{-1}V_r\Sigma V_r^*)^{+}_{NN} \\ &= U_r\Sigma^{-1}U_r^* M U_r\Sigma V_r^* = U_r\Sigma V_r^* N^{-1}V_r\Sigma^{-1}V_r^* \\ &= U_r V_r^*. \end{aligned} \tag{5.3}$$

Thus, the WPD can be computed by using MN-SVD.

5.1. Method based on MN-SVD. The method for computing WPD based on MN-SVD can be described by the following computational procedure:

1. Compute the MN-SVD (2.8) of $A \in \mathbb{C}_r^{m\times n}$, forming only the first r columns U_r and V_r of U and V, respectively;
2. Form G, H, and E according to (5.1), (5.2), and (5.3), respectively.

The algorithm and programs for (2.8) can be obtained according to the proof of Theorem 5.2.2 in [24] (i.e., Lemma 2.1 in this paper) or Theorem 3 in [23] and the algorithm and programs for singular value decomposition (SVD).

The values of the weighted unitary polar factor E and the generalized positive semidefinite polar factors G, H obtained above are accurate. However, it is unnecessary and expensive to calculate their exact values for most applications [9]. The following subsection will develop an alternative method for computing WPD, which is more effective in practice.

5.2. An iterative method. Consider the following iteration:

$$X_0 = A \in \mathbb{C}_r^{m\times n}, \tag{5.4a}$$

$$X_{k+1} = \frac{1}{2}\left(X_k + (X_{k,MN}^{+})^{\#}\right), \quad k = 0, 1, 2, \ldots . \tag{5.4b}$$

We claim that the sequence X_k converges to the weighted unitary polar factor E of A's WPD. To prove this, we make use of MN-SVD introduced in Lemma 2.1. Define

$$D_k = U^* M X_k N^{-1} V. \tag{5.5}$$

Then, from Lemma 2.1 and (5.4), we obtain

$$D_0 = U^* M X_0 N^{-1} V = U^* M A N^{-1} V = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix}, \tag{5.6}$$

$$D_{k+1} = \frac{1}{2}(D_k + U^* M (X_{k,MN}^{+})^{\#} N^{-1} V). \tag{5.7}$$

According to Lemma 2.5, we can get

$$\begin{aligned} U^* M (X_{k,MN}^{+})^{\#} N^{-1} V &= U^* M M^{-1} (X_{k,MN}^{+})^* N N^{-1} V = U^* (X_{k,MN}^{+})^* V \\ &= U^* (N^{-1/2} (M^{1/2} X_k N^{-1/2})^{+} M^{1/2})^* V \\ &= U^* M^{1/2} ((M^{1/2} X_k N^{-1/2})^{+})^* N^{-1/2} V, \end{aligned} \tag{5.8}$$

while

$$\begin{aligned} (M^{-1/2} D_k N^{1/2})_{MN}^{+} &= N^{-1/2} (M^{1/2} M^{-1/2} D_k N^{1/2} N^{-1/2})^{+} M^{1/2} \\ &= N^{-1/2} D_k^{+} M^{1/2}, \end{aligned} \tag{5.9}$$

which implies

$$N^{1/2} (M^{-1/2} D_k N^{1/2})_{MN}^{+} M^{-1/2} = D_k^{+}. \tag{5.10}$$

Meanwhile, according to (5.5), Lemma 2.5, and Lemma 2.4, we have

(5.11)
$$\begin{aligned} N^{1/2}(M^{-1/2}D_kN^{1/2})^+_{MN}M^{-1/2} &= N^{1/2}(M^{-1/2}U^*MX_kN^{-1}VN^{1/2})^+_{MN}M^{-1/2} \\ &= N^{1/2}[N^{-1/2}(M^{1/2}M^{-1/2}U^*MX_kN^{-1}VN^{1/2}N^{-1/2})^+M^{1/2}]M^{-1/2} \\ &= (U^*M^{1/2}M^{1/2}X_kN^{-1/2}N^{-1/2}V)^+ \\ &= (N^{-1/2}V)^*(M^{1/2}X_kN^{-1/2})^+(U^*M^{1/2})^* \\ &= V^*N^{-1/2}(M^{1/2}X_kN^{-1/2})^+M^{1/2}U, \end{aligned}$$

which together with (5.10) and (5.8) gives

$$U^*M(X^+_{k,MN})^{\#}N^{-1}V = (D^+_k)^*. \tag{5.12}$$

Then, (5.7) can be rewritten as

$$D_{k+1} = \frac{1}{2}(D_k + (D^+_k)^*). \tag{5.13}$$

Using the method from [4, 9], we can conclude that

$$D_k \to \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}, \quad k \to \infty. \tag{5.14}$$

Furthermore, by (5.5), we can obtain

$$UD_kV^* = UU^*MX_kN^{-1}VV^* = UU^{-1}X_k(V^*)^{-1}V^* = X_k. \tag{5.15}$$

Therefore,

$$X_k \to U\begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}V^* = U_rV_r^* = E, \quad k \to \infty. \tag{5.16}$$

Then the sequence X_k converges to the weighted unitary polar factor E.

In addition, similarly as in the discussions in [9], we can get

$$D_{k+1} - \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix} = \frac{1}{2}\left(D_k - \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}\right)D^+_k\left(D_k - \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}\right). \tag{5.17}$$

Premultiplying and postmultiplying the equation (5.17) by U and V^*, respectively, and considering (5.15), we have

$$X_{k+1} - E = \frac{1}{2}U\left(D_k - \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}\right)D^+_k\left(D_k - \begin{pmatrix} I_r & 0 \\ 0 & 0 \end{pmatrix}\right)V^*, \tag{5.18}$$

which combined with (2.2) leads to

$$\|X_{k+1}-E\|_{MN} = \left\|\frac{1}{2}U\left(D_k-\begin{pmatrix} I_r & 0\\ 0 & 0\end{pmatrix}\right)D_k^+\left(D_k-\begin{pmatrix} I_r & 0\\ 0 & 0\end{pmatrix}\right)V^*\right\|_{MN} \tag{5.19}$$

$$= \frac{1}{2}\left\|M^{1/2}U\left(D_k-\begin{pmatrix} I_r & 0\\ 0 & 0\end{pmatrix}\right)D_k^+\left(D_k-\begin{pmatrix} I_r & 0\\ 0 & 0\end{pmatrix}\right)V^*N^{-1/2}\right\|_2$$

$$= \frac{1}{2}\left\|\left(D_k-\begin{pmatrix} I_r & 0\\ 0 & 0\end{pmatrix}\right)D_k^+\left(D_k-\begin{pmatrix} I_r & 0\\ 0 & 0\end{pmatrix}\right)\right\|_2$$

$$= \frac{1}{2}\max_{1\le i\le r}\left(\frac{(\sigma_i(X_k)-1)^2}{\sigma_i(X_k)}\right).$$

Similarly,

$$\|X_{k+1}-E\|_{F(MN)} = \frac{1}{2}\left\|\left(D_k-\begin{pmatrix} I_r & 0\\ 0 & 0\end{pmatrix}\right)D_k^+\left(D_k-\begin{pmatrix} I_r & 0\\ 0 & 0\end{pmatrix}\right)\right\|_F \tag{5.20}$$

$$= \frac{1}{2}\left(\sum_{i=1}^{r}\left(\frac{(\sigma_i(X_k)-1)^2}{\sigma_i(X_k)}\right)^2\right)^{1/2},$$

where $\sigma_i(X_k)$ is the nonzero (M,N) weighted singular value of X_k. Thus, we have the following theorem.

THEOREM 5.1. [illegible] $A\in\mathbb{C}_r^{m\times n}$ [illegible] (5.4). [illegible] X_k [illegible] E [illegible] A' [illegible] (5.19) [illegible] (5.20) [illegible].

5.3. Accelerating convergence. Similar to the explanations in [4, 9], the quadratic convergence of iteration (5.4) ensures rapid convergence in the final stages of the iteration. Initially, however, the speed of convergence can be inordinately slow; hence it is necessary to scale the matrix A before the iteration.

5.3.1. The theoretical result. Higham [9] provides a quite good method to scale the matrix A for computing the polar decomposition; the method can be applied to the iteration (5.4).

Consider the scaling $X_k\to\gamma_k X_k, \gamma_k>0$. From (5.4), we have

$$X_0 = A\in\mathbb{C}_r^{m\times n}, \tag{5.21a}$$

$$X_{k+1} = \frac{1}{2}\left(\gamma_k X_k+\frac{1}{\gamma_k}(X_{k,MN}^+)^{\#}\right),\quad k=0,1,2,\ldots. \tag{5.21b}$$

In this case, the equalities (5.19) and (5.20) are changed to

$$\|X_{k+1}-E\|_{MN} = \frac{1}{2}\max_{1\le i\le r}\left(\frac{(\gamma_k\sigma_i(X_k)-1)^2}{\gamma_k\sigma_i(X_k)}\right), \tag{5.22}$$

$$\|X_{k+1}-E\|_{F(MN)} = \frac{1}{2}\left(\sum_{i=1}^{r}\left(\frac{(\gamma_k\sigma_i(X_k)-1)^2}{\gamma_k\sigma_i(X_k)}\right)^2\right)^{1/2}. \tag{5.23}$$

Applying the method introduced in [22], we can show that the optimal parameter is

$$\gamma_{opt}^{(k)} = (\sigma_1(X_k)\sigma_r(X_k))^{-1/2}. \tag{5.24}$$

However, it is not feasible to compute $\gamma_{opt}^{(k)}$ exactly at each stage, since this would require computation of the extremal (M, N) weighted singular values of X_k, but a good approximation to $\gamma_{opt}^{(k)}$ can be computed at negligible cost.

5.3.2. The practical algorithm. Define

$$\alpha_k = \sqrt{\left\|M^{1/2}X_kN^{-1/2}\right\|_1 \left\|M^{1/2}X_kN^{-1/2}\right\|_\infty},$$

$$\beta_k = \sqrt{\left\|N^{1/2}X_{k,MN}^{+}M^{-1/2}\right\|_1 \left\|N^{1/2}X_{k,MN}^{+}M^{-1/2}\right\|_\infty},$$

where $\|X\|_1$ and $\|X\|_\infty$ denote the 1-norm and ∞-norm (see, e.g., [2]) of the matrix X, respectively. According to the introductions in [4, 9], the optimal parameter $\gamma_{opt}^{(k)}$ can be replaced by $\gamma_{pra}^{(k)} = \sqrt{\beta_k/\alpha_k}$. (A detailed proof can be found in [9].) As a result, we have the following practical algorithm.

Given weight matrices M and N;
$X_0 = A \in \mathbb{C}_r^{m\times n}$,
$Y_k = X_{k,MN}^{+}, \quad k = 0, 1, 2, \ldots$.
if $\|X_k - E\|_{MN} \le \varepsilon$ (a very small positive number),
$\gamma_k = 1$,
else
$\alpha_k = \sqrt{\left\|M^{1/2}X_kN^{-1/2}\right\|_1 \left\|M^{1/2}X_kN^{-1/2}\right\|_\infty}$,
$\beta_k = \sqrt{\left\|N^{1/2}Y_kM^{-1/2}\right\|_1 \left\|N^{1/2}Y_kM^{-1/2}\right\|_\infty}$,
$\gamma_k = \sqrt{\frac{\beta_k}{\alpha_k}}$,
$X_{k+1} = \frac{1}{2}\left(\gamma_k X_k + \frac{1}{\gamma_k}Y_k^{\#}\right)$.
Until converged.

Thus, we have $E \approx X = X_{k+1}$. In order to compute the approximate solution of the generalized positive semidefinite polar factor H (G can be discussed similarly; hereafter we study only H) and ensure that it would satisfy $H^{\#} = H$, we take the following matrix K as an AGPSD polar factor of H:

$$K = \frac{1}{2}(X^{\#}A + A^{\#}X). \tag{5.25}$$

In section 6, we will discuss the error bounds for the AGPSD polar factor K.

5.4. Numerical experiment. Next we use a testing matrix from [4] to test the iterative methods given above. The matrix is

$$A = \begin{pmatrix} B + C & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{C}_{35}^{38\times 36},$$

where

$$B = \mathrm{diag}(1, 2^4, 3^4, \cdots, 35^4) \times 10,$$

$$C = \begin{pmatrix} sum(\quad(1,10)) & \cdots & \quad(\quad(1,10)) \\ \vdots & \ddots & \vdots \\ \quad(\quad(1,10)) & \cdots & \quad(\quad(1,10)) \end{pmatrix} \times 10^6 \in \mathbb{C}^{35\times 35},$$

and the stopping criterion is

$$\|X_{k+1} - E\|_{MN} \leq 2 \times 10^{-8}, \tag{5.26}$$

in which the weighted unitary polar factor E is computed by MN-SVD.

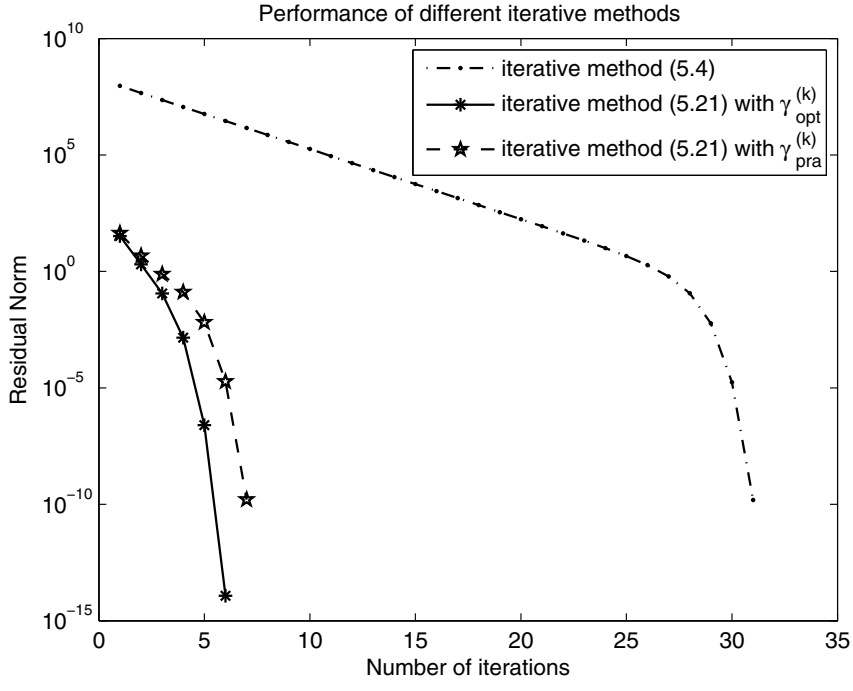


FIG. 5.1. *Performance of different iterative methods.*

Moreover, in order to obtain the reasonable weight matrices, a method to derive the Hermitian positive definite matrix is introduced as follows.

1. Generate an $m \times m$ random array with normal distribution: $M_1 = \quad(m, m)$;
2. Compute the covariance matrix of the random array: $M_2 = cov(M_1)$;
3. Form the $m \times m$ Hermitian positive definite matrix: $M = M_2 + eye(m) * m$; where the commands can be found in detail in any book on MATLAB.

Figure 5.1 describes the comparisons among the above iterative methods. According to the stopping criteria (5.26), the iterative method (5.4) needs to iterate 31 times, while the iterative method (5.21) with $\gamma_{opt}^{(k)}$ needs 6 times and the iterative method (5.21) with $\gamma_{pra}^{(k)}$ needs 7 times.

6. Error bounds for AGPSD polar factor. In this section, we will present the error bounds for the AGPSD polar factor K defined in (5.25) in weighted unitary invariant norm and weighted Frobenius norm, respectively.

THEOREM 6.1. Let $A \in \mathbb{C}_r^{m\times n}$ [illegible] 3.8. [illegible] $X \in \mathbb{C}^{m\times n}$ [illegible] E [illegible] K [illegible] (5.25), [illegible]

$$\|H-K\|_{(NN)} \le \|A\|_{MN}\,\|X-E\|_{(MN)}. \tag{6.1}$$

[illegible]. By (5.2), (5.3), and (2.10), we can obtain

$$2H = E^{\#}A + A^{\#}E,$$

which combined with (5.25) implies

$$\begin{aligned} H-K &= \frac{1}{2}(2H - X^{\#}A - A^{\#}X) = \frac{1}{2}(E^{\#}A + A^{\#}E - X^{\#}A - A^{\#}X) \\ &= \frac{1}{2}\left((E^{\#}-X^{\#})A + A^{\#}(E-X)\right), \end{aligned} \tag{6.2}$$

which together with (2.4) and (2.2) gives

$$\begin{aligned} \|H-K\|_{(NN)} &= \frac{1}{2}\left\|(E^{\#}-X^{\#})A + A^{\#}(E-X)\right\|_{(NN)} \\ &= \frac{1}{2}\left\|N^{1/2}\left((E^{\#}-X^{\#})A + A^{\#}(E-X)\right)N^{-1/2}\right\| \\ &\le \frac{1}{2}\left(\left\|N^{1/2}(E^{\#}-X^{\#})AN^{-1/2}\right\| + \left\|N^{1/2}A^{\#}(E-X)N^{-1/2}\right\|\right) \\ &= \frac{1}{2}\left\|N^{-1/2}(E-X)^{*}M^{1/2}M^{1/2}AN^{-1/2}\right\| \\ &\quad + \frac{1}{2}\left\|N^{-1/2}A^{*}M^{1/2}M^{1/2}(E-X)N^{-1/2}\right\| \\ &\le \frac{1}{2}\left\|N^{-1/2}(E-X)^{*}M^{1/2}\right\|\left\|M^{1/2}AN^{-1/2}\right\|_2 \\ &\quad + \frac{1}{2}\left\|N^{-1/2}A^{*}M^{1/2}\right\|_2\left\|M^{1/2}(E-X)N^{-1/2}\right\| \\ &= \|A\|_{MN}\,\|E-X\|_{(MN)} = \|A\|_{MN}\,\|E-X\|_{(MN)}. \end{aligned} \tag{6.3}$$

□

THEOREM 6.2. Let $A \in \mathbb{C}_r^{m\times n}$ [illegible] 3.8. [illegible] $X \in \mathbb{C}^{m\times n}$ [illegible] E [illegible] $R(X^{\#}) = R(A^{\#})$ [illegible] K [illegible] (5.25), [illegible]

$$\begin{aligned} \|H-K\|_{F(NN)} &\le \frac{1}{2}\|A\|_{MN}\left\|(E-X)^{\#}(E-X)\right\|_{F(NN)} \\ &\quad + \frac{1}{2}\left(\|A\|_{MN} - \frac{1}{\left\|A_{MN}^{+}\right\|_{NM}}\right)\sqrt{\|X^{\#}(E-X)\|_{F(NN)}^2 - \frac{1}{r}\left|\mathrm{tr}[X^{\#}(E-X)]\right|^2}. \end{aligned} \tag{6.4}$$

. From Theorem 3.8 and (5.2), we can obtain

$$R(A^{\#}) = R(E^{\#}) = R(H) = R(N^{-1}V_r).$$

Then

$$R(X^{\#}) = R(A^{\#}) = R(E^{\#}) = R(H) = R(N^{-1}V_r),$$

which combined with Lemma 3.3 and (5.3) leads to

$$X^{\#}X = E^{\#}E = N^{-1}V_rV_r^*. \tag{6.5}$$

According to (6.2) and (3.12), and observing the fact that $H^{\#} = H$, we have

$$\begin{aligned} H - K &= \frac{1}{2}(E^{\#}A + A^{\#}E - X^{\#}A - A^{\#}X) \\ &= \frac{1}{2}(E^{\#}EH + HE^{\#}E - X^{\#}EH - HE^{\#}X) \\ &= \frac{1}{2}H(E^{\#}E - E^{\#}X - X^{\#}E + X^{\#}X) \\ &\quad + \frac{1}{2}(HX^{\#}E - HX^{\#}X + E^{\#}EH - X^{\#}EH) \\ &= \frac{1}{2}\left(H(E-X)^{\#}(E-X) + HX^{\#}(E-X) + (E^{\#} - X^{\#})EH\right), \end{aligned} \tag{6.6}$$

whereas if $(E^{\#} - X^{\#})E = E^{\#}E - X^{\#}E = X^{\#}X - X^{\#}E = X^{\#}(X - E)$, then

$$\begin{aligned} &\|H - K\|_{F(NN)} \\ &= \frac{1}{2}\left\|H(E-X)^{\#}(E-X) + HX^{\#}(E-X) - X^{\#}(E-X)H\right\|_{F(NN)} \\ &\le \frac{1}{2}\left\|N^{1/2}H(E-X)^{\#}(E-X)N^{-1/2}\right\|_F \\ &\quad + \frac{1}{2}\left\|N^{1/2}(HX^{\#}(E-X) - X^{\#}(E-X)H)N^{-1/2}\right\|_F \\ &\le \frac{1}{2}\|H\|_{NN}\left\|(E-X)^{\#}(E-X)\right\|_{F(NN)} \\ &\quad + \frac{1}{2}\left\|N^{1/2}(HX^{\#}(E-X) - X^{\#}(E-X)H)N^{-1/2}\right\|_F. \end{aligned} \tag{6.7}$$

However, considering (5.2) and the fact that $V^*N^{-1}V = (N^{-1/2}V)^*N^{-1/2}V = I_n$, we have

$$\begin{aligned} &\left\|N^{1/2}(HX^{\#}(E-X) - X^{\#}(E-X)H)N^{-1/2}\right\|_F \\ &= \left\|N^{1/2}(N^{-1}V_r\Sigma V_r^*X^{\#}(E-X) - X^{\#}(E-X)N^{-1}V_r\Sigma V_r^*)N^{-1/2}\right\|_F \\ &= \left\|N^{-1/2}V_r\Sigma V_r^*X^{\#}(E-X)N^{-1/2} - N^{1/2}X^{\#}(E-X)N^{-1}V_r\Sigma V_r^*N^{-1/2}\right\|_F \\ &= \left\|V^*N^{-1}V_r\Sigma V_r^*X^{\#}(E-X)N^{-1}V - V^*X^{\#}(E-X)N^{-1}V_r\Sigma V_r^*N^{-1}V\right\|_F, \end{aligned}$$

which together with $V_r^* N^{-1} V_r = I_r$, $V_{n-r}^* N^{-1} V_r = V_r^* N^{-1} V_{n-r} = 0$, (5.3), and (6.5) gives

$$\begin{aligned}(6.8)\quad &\left\| N^{1/2}(HX^{\#}(E-X) - X^{\#}(E-X)H)N^{-1/2} \right\|_F \\ &\qquad = \left\| \Sigma V_r^* X^{\#}(E-X)N^{-1}V_r - V_r^* X^{\#}(E-X)N^{-1}V_r \Sigma \right\|_F .\end{aligned}$$

Let $C = V_r^* X^{\#}(E-X)N^{-1}V_r = (c_{ij})$. Then (6.8) can be rewritten as

$$\begin{aligned}(6.9)\quad &\left\| N^{1/2}(HX^{\#}(E-X) - X^{\#}(E-X)H)N^{-1/2} \right\|_F^2 = \|\Sigma C - C\Sigma\|_F^2 \\ &\qquad = \sum_{i,j=1}^{r} ((\sigma_i - \sigma_j)|c_{ij}|)^2 \le (\sigma_1 - \sigma_r)^2 \left(\|C\|_F^2 - \sum_{i=1}^{r} (|c_{ii}|)^2 \right).\end{aligned}$$

Since $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ and $N^{-1}V_r V_r^* X^{\#} = X^{\#}, N^{-1}V_r V_r^* E^{\#} = E^{\#}$, we have

$$\begin{aligned}(6.10)\quad &\sum_{i=1}^{r} (|c_{ii}|)^2 \ge \frac{1}{r} \left| \sum_{i=1}^{r} c_{ii} \right|^2 = \frac{1}{r} |\operatorname{tr}(C)|^2 = \frac{1}{r} \left| \operatorname{tr}(V_r^* X^{\#}(E-X)N^{-1}V_r) \right|^2 \\ &\qquad = \frac{1}{r} \left| \operatorname{tr}(N^{-1}V_r V_r^* X^{\#}(E-X)) \right|^2 = \frac{1}{r} \left| \operatorname{tr}(X^{\#}(E-X)) \right|^2\end{aligned}$$

and

$$\begin{aligned}(6.11)\quad \|C\|_F^2 &= \operatorname{tr}(C^*C) = \operatorname{tr}\left((V_r^* X^{\#}(E-X)N^{-1}V_r)^* V_r^* X^{\#}(E-X)N^{-1}V_r \right) \\ &= \operatorname{tr}\left(V_r^* N^{-1}(E-X)^*(X^{\#})^* V_r V_r^* X^{\#}(E-X)N^{-1}V_r \right) \\ &= \operatorname{tr}\left(N^{-1}V_r V_r^* N^{-1}(E-X)^* M X N^{-1} V_r V_r^* X^{\#}(E-X) \right) \\ &= \operatorname{tr}\left(N^{-1}V_r V_r^* (E-X)^{\#} X X^{\#}(E-X) \right) \\ &= \operatorname{tr}\left((E-X)^{\#} X X^{\#}(E-X) \right).\end{aligned}$$

Then, observing (2.5) and (6.11), we have

$$(6.12)\qquad \|C\|_F^2 = \left\| X^{\#}(E-X) \right\|_{F(NN)}^2 .$$

Therefore, together with (6.12), (6.10), (6.9), and (6.7), we can conclude that

$$\begin{aligned}(6.13)\quad \|H-K\|_{F(NN)} &\le \frac{1}{2} \|H\|_{NN} \left\| (E-X)^{\#}(E-X) \right\|_{F(NN)} \\ &\quad + \frac{1}{2}(\sigma_1 - \sigma_r) \sqrt{ \|X^{\#}(E-X)\|_{F(NN)}^2 - \frac{1}{r} |\operatorname{tr}(X^{\#}(E-X))|^2 }.\end{aligned}$$

The desired result follows from $\|A\|_{MN} = \|H\|_{NN} = \sigma_1$ and $\left\| A_{MN}^{+} \right\|_{MN} = \frac{1}{\sigma_r}$. ∎

If $R = A - XK$ is defined as the residual of A about X, then we have the following theorem.

THEOREM 6.3. [illegible] 6.2 [illegible]

$$(6.14)\qquad \|H-K\|_{F(NN)} \le \sqrt{ \|A\|_{MN}^2 \|X^{\#}(X-E)\|_{F(NN)}^2 - \|X^{\#}R\|_{F(NN)}^2 }.$$

. According to (5.25) and the fact that $X^{\#}XH = HX^{\#}X = H$, we have

$$H - K = \frac{1}{2}(X^{\#}XH + HX^{\#}X - X^{\#}EH - HE^{\#}X) \tag{6.15}$$
$$= \frac{1}{2}\left(H(X-E)^{\#}X + X^{\#}(X-E)H\right).$$

Meanwhile, from $R = A - XK$, we can get

$$X^{\#}R = X^{\#}EH - \frac{1}{2}X^{\#}X(X^{\#}EH + HE^{\#}X) \tag{6.16}$$
$$= \frac{1}{2}(2X^{\#}EH - X^{\#}XX^{\#}EH - X^{\#}XHE^{\#}X)$$
$$= \frac{1}{2}(X^{\#}EH - HE^{\#}X) = \frac{1}{2}(X^{\#}EH - H + H - HE^{\#}X)$$
$$= \frac{1}{2}(X^{\#}EH - X^{\#}XH + HX^{\#}X - HE^{\#}X)$$
$$= \frac{1}{2}\left(H(X^{\#} - E^{\#})X - X^{\#}(X-E)H\right).$$

In addition, we have the following fact:

$$\|Z\|^2_{F(NN)} = \left\|N^{1/2}ZN^{-1/2}\right\|_F^2 = \left\|\frac{1}{2}(N^{1/2}ZN^{-1/2} + N^{-1/2}Z^*N^{1/2})\right\|_F^2$$
$$+ \left\|\frac{1}{2}(N^{1/2}ZN^{-1/2} - N^{-1/2}Z^*N^{1/2})\right\|_F^2.$$

Setting $Z = X^{\#}(X-E)H$ and considering (6.15), (6.16), we have

$$\|H-K\|^2_{F(NN)} = \left\|\frac{1}{2}(Z^{\#} + Z)\right\|^2_{F(NN)} = \left\|\frac{1}{2}(N^{1/2}(Z^{\#} + Z)N^{-1/2})\right\|_F^2 \tag{6.17}$$
$$= \left\|\frac{1}{2}(N^{-1/2}Z^*N^{1/2} + N^{1/2}ZN^{-1/2})\right\|_F^2$$
$$= \|Z\|^2_{F(NN)} - \left\|\frac{1}{2}(N^{-1/2}Z^*N^{1/2} - N^{1/2}ZN^{-1/2})\right\|_F^2$$
$$= \|Z\|^2_{F(NN)} - \left\|X^{\#}R\right\|^2_{F(NN)},$$

i.e.,

$$\|H-K\|^2_{F(NN)} = \left\|X^{\#}(X-E)H\right\|^2_{F(NN)} - \left\|X^{\#}R\right\|^2_{F(NN)} \tag{6.18}$$
$$\leq \|H\|^2_{NN}\left\|X^{\#}(X-E)\right\|^2_{F(NN)} - \left\|X^{\#}R\right\|^2_{F(NN)}.$$

Then, the proof is completed by noting that $\|A\|_{MN} = \|H\|_{NN} = \sigma_1$. ∎

7. Conclusions. In this paper, we prove the unique WPD theorem and derive some necessary and sufficient conditions for SWPD. A new partial ordering of matrices called WGL partial ordering is also defined and studied. Using the new partial ordering, we present a characteristic of SWPD. In addition, the methods for computing WPD and the error bounds for the AGPSD polar factor are also discussed. However, this paper is not involved with various applications of WPD. As we all know, the (generalized) polar decomposition has a wide range of applications. Therefore, it is of interest to probe the applications of WPD with the development of research on the weighted generalized inverses of matrices, weighted least squares problems, weighted optimization problem, and so on. This will be our future work.

Acknowledgments. The authors would like to thank the editor, Professor R.-C. Li, and two anonymous referees for their valuable comments and helpful suggestions, which improved the presentation. The second author also wishes to thank Professor J. Hauke for sending him reference [7].

REFERENCES

[1] J. K. Baksalary, E. P. Liski, and G. Trenkler, *Mean square error matrix improvements and admissibility of linear estimators*, J. Statist. Plann. Inference, 23 (1989), pp. 313–325.
[2] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed., Springer, New York, 2003.
[3] X.-S. Chen and W. Li, *Error bounds for the approximate generalized polar factors*, Acta Math. Appl. Sin., 29 (2006), pp. 270–275 (in Chinese).
[4] K. Du, *The iterative methods for computing the polar decomposition of rank-deficient matrix*, Appl. Math. Comput., 162 (2005), pp. 95–102.
[5] W. Gander, *Algorithms for the polar decomposition*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 1102–1115.
[6] J. Gross, J. Hauke, and A. Markiewicz, *Partial orderings, preorderings, and the polar decomposition of matrices*, Linear Algebra Appl., 289 (1999), pp. 161–168.
[7] J. Hauke and A. Markiewicz, *On orderings induced by the Löwner partial ordering*, Appl. Math. (Warsaw), 22 (1994), pp. 145–154.
[8] J. Hauke and A. Markiewicz, *On partial orderings on the set of rectangular matrices*, Linear Algebra Appl., 219 (1995), pp. 187–193.
[9] N. J. Higham, *Computing the polar decomposition—with applications*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1160–1174.
[10] N. J. Higham, D. S. Mackey, N. Mackey, and F. Tisseur, *Computing the polar decomposition and the matrix sign decomposition in matrix groups*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1178–1192.
[11] N. J. Higham and P. Papadimitriou, *A parallel algorithm for computing the polar decomposition*, Parallel Comput., 20 (1994), pp. 1161–1173.
[12] N. J. Higham and R. S. Schreiber, *Fast polar decomposition of an arbitrary matrix*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 648–655.
[13] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[14] C. Kenney and A. J. Laub, *Polar decomposition and matrix sign function condition estimates*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 488–504.
[15] C. Kenney and A. J. Laub, *On scaling Newton's method for polar decomposition and the matrix sign function*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 688–706.
[16] X.-J. Liu, *Partial Orderings and Generalized Inverse of Matrices*, Ph.D. thesis, Xidian University, Xi'an, China, 2003 (in Chinese).
[17] X.-J. Liu and H. He, *On weighted partial isometric matrices*, J. Univ. Sci. Technol. Suzhou (Natur. Sci.), 20 (2003), pp. 22–24 (in Chinese).
[18] A. Markiewicz, *Simultaneous polar decomposition of rectangular complex matrices*, Linear Algebra Appl., 289 (1999), pp. 279–284.
[19] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and Its Applications*, Wiley, New York, 1971.

[20] C. R. Rao and M. B. Rao, *Matrix Algebra and Its Applications to Statistics and Econometrics*, World Scientific, Hong Kong, 1998.
[21] J.-G. Sun, *On the approximate polar factors*, Math. Numer. Sin., 1 (1991), pp. 45–50 (in Chinese).
[22] J.-G. Sun and C.-H. Chen, *Generalized polar decomposition*, Math. Numer. Sin., 11 (1989), pp. 262–273 (in Chinese).
[23] C. F. Van Loan, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal., 13 (1976), pp. 76–83.
[24] G.-R. Wang, Y.-M. Wei, and S.-Z. Qiao, *Generalized Inverses: Theory and Computations*, Science Press, Beijing, 2004.
[25] W.-J. Zhuang, *Polar decomposition and GL partial ordering for quaternion rectangular matrices*, Adv. Math. (China), 34 (2005), pp. 187–193 (in Chinese).

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 2, pp. 925–938

CONVERGENCE OF STATIONARY ITERATIVE METHODS FOR HERMITIAN SEMIDEFINITE LINEAR SYSTEMS AND APPLICATIONS TO SCHWARZ METHODS*

ANDREAS FROMMER†, REINHARD NABBEN‡, AND DANIEL B. SZYLD§

Abstract. A simple proof is presented of a quite general theorem on the convergence of stationary iterations for solving singular linear systems whose coefficient matrix is Hermitian and positive semidefinite. In this manner, elegant proofs are obtained of some known convergence results, including the necessity of the P-regular splitting result due to Keller, as well as recent results involving generalized inverses. Other generalizations are also presented. These results are then used to analyze the convergence of several versions of algebraic additive and multiplicative Schwarz methods for Hermitian positive semidefinite systems.

Key words. linear systems, Hermitian semidefinite systems, singular systems, stationary iterative methods, seminorm, convergence analysis, algebraic Schwarz methods

AMS subject classifications. 65F10, 65F20

DOI. 10.1137/080714038

1. Introduction.

We consider the linear system

$$Ax = b, \tag{1.1}$$

where the coefficient matrix $A \in \mathbb{C}^{n\times n}$ is assumed to be singular and Hermitian positive semidefinite. Denoting by $\mathrm{Null}(A)$ the nullspace of A and by $\mathrm{Range}(A)$ its range, we assume that $b \in \mathrm{Range}(A)$. This implies that the solution set of (1.1) is nonempty and it is given as an affine space $x^* + \mathrm{Null}(A)$ for some $x^* \in \mathbb{C}^n$ solution of (1.1).

If A is large and sparse, iterative methods for solving (1.1) are the standard approach. In this paper, we focus on stationary iterative methods, including, for example, certain algebraic multigrid methods and additive and multiplicative Schwarz methods. Sometimes, these iterations are accelerated by using them as preconditioners to Krylov subspace methods like conjugate gradients. While we do not consider the latter aspect in any detail in this work, let us just mention that one usually assumes convergence of the preconditioner as a prerequisite in this context, so our work is relevant in this case as well.

We consider the very general situation in which we are given an iteration matrix H for (1.1) of the form

$$H = I - \widetilde{M}A, \tag{1.2}$$

*Received by the editors January 23, 2008; accepted for publication (in revised form) by M. Benzi April 14, 2008; published electronically September 25, 2008.
http://www.siam.org/journals/simax/30-2/71403.html

†Fachbereich Mathematik und Naturwissenschaften, Universität Wuppertal, Gauß-Straße 20, D-42097 Wuppertal, Germany (frommer@math.uni-wuppertal.de).

‡Institut für Mathematik, MA 3-3, Technische Universität Berlin, Straße des 17. Juni 136, D-10623 Berlin, Germany (nabben@math.tu-berlin.de).

§Department of Mathematics, Temple University (038-16), 1805 N. Broad Street, Philadelphia, PA 19122-6094 (szyld@temple.edu). This author's research was supported in part by U.S. National Science Foundation grant CCF-0514889 and U.S. Department of Energy grant DE-FG02-05ER25672.

where $\widetilde{M} \in \mathbb{C}^{n\times n}$ is a matrix which might be singular but is injective on Range(A), i.e.,

$$\text{Null}(\widetilde{M}A) = \text{Null}(A). \tag{1.3}$$

The matrices H and $\widetilde{M}$ induce the iteration

$$x^{k+1} = Hx^k + \widetilde{M}b. \tag{1.4}$$

Since any solution x^* of (1.1) satisfies $\widetilde{M}Ax^* = \widetilde{M}b$, we see that each such x^* is a fixed point of the iteration (1.4). Conversely, if x^* is a fixed point of (1.4), then $0 = -\widetilde{M}Ax^* + \widetilde{M}b$, and since $\widetilde{M}$ is injective on Range(A) we get $Ax^* = b$. We conclude that under the conditions (1.2) and (1.3), x^* is a solution of (1.1) if and only if x^* is a fixed point of (1.4).

The rest of this paper is devoted to the analysis of situations where we can guarantee that the iteration (1.4) converges to a fixed point. Due to the singularity of A, such a limiting fixed point usually depends on the starting vector x^0. Actually, condition (1.3) implies that convergence of the iteration (1.4) is equivalent to H being semiconvergent according to the following definition;[1] see, e.g., [3], [7], [18].

DEFINITION 1.1. [illegible] $H \in \mathbb{C}^{n\times n}$ [illegible] $\rho(H) = 1$, $\lambda = 1$ [illegible] 1 [illegible] $\lambda = 1$ [illegible] H [illegible]

It follows, then, that one goal is to find simple conditions for which we can show that H of the form (1.2) is such that (1.3) holds and it is semiconvergent.

Our general form of the iteration operator from (1.2) applies in particular to iterations induced by splittings of the form $A = M - N$, M nonsingular, in which $\widetilde{M}$ is taken to be M^{-1}. Then condition (1.3) is automatically satisfied. There are iterations which can be interpreted as being of the form (1.2) with $\widetilde{M} = M^\dagger$, the Moore–Penrose pseudoinverse of some singular matrix M; see [7], [12], [13], where such iterations are studied. This situation occurs in particular in the analysis of Schwarz iterations where the artificial boundary conditions between subdomains are of Neumann type; see, e.g., [17], [19].

The rest of the paper is organized as follows. In section 2 we derive a fundamental convergence result based on an estimate in the energy seminorm. In section 3 the fundamental result is used in two directions: We obtain simple and elegant proofs for some known convergence results, and we develop new convergence results which improve over some that have been published previously. We then consider algebraic additive and multiplicative Schwarz methods. The paper finishes with a conclusion in section 4. We mention that applications of the fundamental result to algebraic multigrid methods are presented in the forthcoming paper [8].

2. A fundamental result. In the analysis to follow, we use the bilinear form $\langle\cdot,\cdot\rangle_A$ defined for a Hermitian matrix $A \in \mathbb{C}^{n\times n}$ as

$$\langle\cdot,\cdot\rangle_A : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C},\ \ (x,y) \mapsto \langle x,y\rangle_A = \langle Ax,y\rangle\ (= \langle x,Ay\rangle).$$

Here, $\langle x,y\rangle$ denotes the standard Euclidean inner product. Since in our context A is only positive semidefinite, the bilinear form is only semidefinite as well. We collect some trivial properties of $\langle\cdot,\cdot\rangle_A$ in the following lemma.

LEMMA 2.1. [illegible] A [illegible]

[1] We note that in some papers such a matrix is simply called *convergent*.

(i) [illegible] $x \in \mathbb{C}^n$ [illegible] $\langle x, x\rangle_A \geq 0$
(ii) $\langle x, x\rangle_A = 0$ [illegible] $x \in \mathrm{Null}(A)$
(iii) [illegible] $x \in \mathrm{Null}(A)$ [illegible] $y \in \mathrm{Null}(A)$ [illegible] $\langle x, y\rangle_A = 0$

In what follows, $\|x\|_A$ denotes the seminorm $\langle x, x\rangle_A^{1/2}$.

We now turn to formulate a fundamental theorem on the convergence of the iteration (1.4). We include a simple proof before discussing how it is related to similar results in recent publications.

THEOREM 2.2. [illegible] $H = I - \widetilde{M}A \in \mathbb{C}^{n\times n}$ [illegible] (1.4) [illegible]

$$x \notin \mathrm{Null}(A) \Longrightarrow \|Hx\|_A < \|x\|_A. \tag{2.1}$$

[illegible]
(i) $\mathrm{Null}(\widetilde{M}A) = \mathrm{Null}(A)$ [illegible] $\widetilde{M}$ [illegible] $\mathrm{Range}(A)$
(ii) H [illegible]

[illegible] $b \in \mathrm{Range}(A)$ [illegible] (1.4) [illegible] (1.1) [illegible] x^0 [illegible]. First observe that $\mathrm{Null}(\widetilde{M}A) = \mathrm{Null}(I - H)$. For $y \notin \mathrm{Null}(A)$ the hypothesis (2.1) gives $Hy \neq y$, i.e., $y \notin \mathrm{Null}(I - H)$. On the other hand $y \in \mathrm{Null}(A)$ implies $y \in \mathrm{Null}(I - H)$, by the definition of H. This shows $\mathrm{Null}(\widetilde{M}A) = \mathrm{Null}(I - H) = \mathrm{Null}(A)$, i.e., (i) holds.

To prove (ii), let x be an eigenvector for an eigenvalue λ of H. If $x \notin \mathrm{Null}(A)$, we have $\|x\|_A > 0$, and from (2.1) we get $|\lambda| \cdot \|x\|_A < \|x\|_A$ which implies $|\lambda| < 1$. If $x \in \mathrm{Null}(A)$, we know that $Hx = x$, i.e., $\lambda = 1$. So $\rho(H) = 1$, and $\lambda = 1$ is the only eigenvalue of modulus 1. It remains to show that $\lambda = 1$ is semisimple.

Assume that, on the contrary, $\lambda = 1$ is not a semisimple eigenvalue of H. Then there exists a level-2 generalized eigenvector for the eigenvalue $\lambda = 1$, i.e., a vector $v \neq 0$ satisfying

$$Hv = v + u, \text{ where } Hu = u, u \neq 0.$$

Since v is not an eigenvector of H we have $v \notin \mathrm{Null}(A)$. We also have $u \in \mathrm{Null}(A)$, since u is an eigenvector of H for the eigenvalue $\lambda = 1$ and $\widetilde{M}$ is injective on $\mathrm{Range}(A)$. Thus, using parts (ii) and (iii) of Lemma 2.1, we get

$$\langle Hv, Hv\rangle_A = \langle v, v\rangle_A + \langle v, u\rangle_A + \langle u, v\rangle_A + \langle u, u\rangle_A = \langle v, v\rangle_A,$$

which contradicts (2.1). Therefore, there is no level-2 generalized eigenvector for the eigenvalue $\lambda = 1$; i.e., $\lambda = 1$ is semisimple. □

[illegible] 2.3. Since $Hx = x$ for $x \in \mathrm{Null}(A)$, the operator H canonically induces a linear operator $\mathcal{H}$ on the quotient space $\mathcal{Q} = \mathbb{C}^n/\mathrm{Null}(A)$ on which $\|\cdot\|_A$ canonically induces a true norm $\|x + \mathrm{Null}(A)\|_A := \|x\|_A$. Therefore, the implication (2.1) is actually equivalent to

$$\|Hx\|_A \leq \|\mathcal{H}\|_A \cdot \|x\|_A \text{ with } \|\mathcal{H}\|_A < 1. \tag{2.2}$$

We will write $\|H\|_A$ for $\|\mathcal{H}\|_A$ in what follows and, for simplicity, we will always formulate our convergence results to come by stating that $\|H\|_A < 1$, having in mind that this means that $\widetilde{M}$ is injective on $\mathrm{Range}(A)$ and that the iteration (1.4) converges to a solution of (1.1) whenever $b \in \mathrm{Range}(A)$.

Theorem 2.2 complements several recently published results. In [12], it was observed that $\|H\|_A < 1$ is sufficient for $\lim_{k\to\infty}(Ax^k - b) = 0$ for the iterates of (1.4) in the case that $\widetilde{M} = M^\dagger$, the Moore–Penrose inverse of a matrix M satisfying Range$(A) \subseteq$ Range(M) and $b \in$ Range(A). This kind of convergence is called [illegible], and an iterative scheme (1.4) satisfying $\|H\|_A < 1$ is called [illegible] (or [illegible]) in [12]. It was then shown in [7] that quotient convergence is actually equivalent to "usual" convergence, i.e., $\lim_{k\to\infty} x^k = x^*$ with $Ax^* = b$. This is precisely the assertion of Theorem 2.2 except that we do not require $\widetilde{M}$ to be a Moore–Penrose pseudoinverse. The references [7], [12] use such pseudoinverses since they view the iteration (1.4) as arising from a splitting $A = M - N$ of A. Since every matrix $\widetilde{M}$ is the Moore–Penrose inverse of its own Moore–Penrose inverse, i.e., $\widetilde{M} = (\widetilde{M}^\dagger)^\dagger$, we see that there is nothing special in requiring $\widetilde{M}$ to be a Moore–Penrose pseudoinverse.

The crucial condition is (2.1) (or, equivalently, (2.2)), implying (1.3) and the semiconvergence of H, which is equivalent to convergence and quotient convergence [7]. The beauty of Theorem 2.2 is that, on one hand, only the condition (2.1) is required for convergence and, on the other, its proof is simple.

3. Applications of the fundamental result. As first applications of Theorem 2.2 we give simple proofs of the necessity of a well-known result of Keller [11] (see also [7]) and a generalization which contains as a special case a recent result from [13].

THEOREM 3.1. [illegible] $A \in \mathbb{C}^{n\times n}$ [illegible] M [illegible] $H = I - M^{-1}A$ [illegible] $M + M^H - A$ [illegible] Range$(M^{-1}A)$ [illegible] $\|H\|_A < 1$

Here, M^H denotes the conjugate transpose of the matrix M.

[illegible]. Using the identity

$$H^H A H = A - AM^{-H}(M + M^H - A)M^{-1}A,$$

we see that

$$\langle Hx, Hx\rangle_A = x^H H^H A H x = \langle x, x\rangle_A - \langle M^{-1}Ax, (M + M^H - A)M^{-1}Ax\rangle. \tag{3.1}$$

For $x \notin$ Null(A) the vector $M^{-1}Ax$ is nonzero, so that due to the positive definiteness of $M + M^H - A$ on Range$(M^{-1}A)$ we obtain

$$x \notin \text{Null}(A) \Longrightarrow \langle Hx, Hx\rangle_A < \langle x, x\rangle_A,$$

and $\|H\|_A < 1$ follows by Remark 2.3.

On the other hand, if $\|H\|_A < 1$, then $\langle Hx, Hx\rangle_A < \langle x, x\rangle_A$ for all $x \notin$ Null(A), so that (3.1) gives

$$\langle M^{-1}Ax, (M + M^H - A)M^{-1}Ax\rangle = \langle x, x\rangle_A - \langle Hx, Hx\rangle_A > 0.$$

Since every nonzero $y \in$ Range$(M^{-1}A)$ can be expressed as $y = M^{-1}Ax$ with $x \notin$ Null(A), this shows that $M + M^H - A$ is positive definite on Range$(M^{-1}A)$. □

Recall that by Theorem 2.2 and Remark 2.3, $\|H\|_A < 1$ implies that the iteration (1.4) converges towards a solution of (1.1) for every starting vector. It is in these terms that the above theorem was originally formulated in [11].

One application of Theorem 3.1 is for the relaxed Gauss–Seidel iteration. With $A = D - L - L^H$ denoting the canonical decomposition of A into its diagonal part

D, its lower triangular part $-L$ and its upper triangular part $-L^H$, one then has $M = \frac{1}{\omega}D - L$. This matrix M is nonsingular if no diagonal element of A is zero, and $M + M^H - A = \frac{2-\omega}{\omega}D$ is positive definite on the whole space for $\omega \in (0, 2)$.

We now turn to the announced generalization, where M is allowed to be singular.

THEOREM 3.2. [illegible] $A \in \mathbb{C}^{n\times n}$ [illegible] $M, \widetilde{M} \in \mathbb{C}^{n\times n}$ [illegible]

$$M\widetilde{M}A = A \tag{3.2}$$

[illegible] $H = I - \widetilde{M}A$ [illegible] $\|H\|_A < 1$, [illegible] $M + M^H - A$, [illegible] Range$(\widetilde{M}A)$

[illegible]. We first observe that since $M\widetilde{M}A = A$ we have $\mathrm{Null}(\widetilde{M}A) = \mathrm{Null}(A)$, i.e., $\widetilde{M}$ is injective on Range(A). We also have $A(\widetilde{M})^H M^H = A$ and thus

$$\begin{aligned} H^H A H &= A - A\widetilde{M}A - A(\widetilde{M})^H A + A(\widetilde{M})^H A\widetilde{M}A \\ &= A - A(\widetilde{M})^H \cdot \big(M + M^H - A\big) \cdot \widetilde{M}A. \end{aligned} \tag{3.3}$$

So, if $M+M^H-A$ is positive definite on Range$(\widetilde{M}A)$, we see that for $x \notin \mathrm{Null}(\widetilde{M}A) = \mathrm{Null}(A)$ one has

$$\|Hx\|_A < \|x\|_A$$

so that $\|H\|_A < 1$ follows again from Remark 2.3.

The converse follows in the same manner as in the proof of Theorem 3.1, so we do not reproduce it here. □

This result allows us to use for $\widetilde{M}$ various generalized inverses of M. In the case that $\widetilde{M}$ is the Moore–Penrose inverse $M^\dagger$ of M, a sufficient condition for $M\widetilde{M}A = A$ is to require Range$(A) \subseteq$ Range(M). With this more restrictive condition, Theorem 3.2 was essentially proved in [12, Theorem 4.4]; see also [7]. The paper [13] uses the same condition Range$(A) \subseteq$ Range(M) but allows $\widetilde{M}$ to just be an inner inverse of M, i.e., an operator satisfying $M\widetilde{M}M = M$. The convergence results there, however, come in a quite different flavor. Instead of assuming the positive definiteness of $M + M^H - A$ on Range$(\widetilde{M}A)$, they require a further, more indirectly defined matrix to be an inner inverse.

We note that Cao [7] presents the following example indicating that condition (3.2) is essential for the necessity part of Theorem 3.2.

[illegible] 3.3. Let

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad M = \frac{1}{2}\begin{bmatrix} 0 & -1 \\ 0 & 1 \end{bmatrix}, \quad \widetilde{M} = M^\dagger = \begin{bmatrix} 0 & 0 \\ -1 & 1 \end{bmatrix},$$

for which it holds that $\widetilde{M}A = A$, and thus $M\widetilde{M}A = MA = M \neq A$. On the other hand, we have that

$$H = I - \widetilde{M}A = I - A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

so that $\|H\|_A = 0 < 1$, but it holds that $(\widetilde{M}A)^H(M^H + M - A)(\widetilde{M}A) = 0$, and thus $M^H + M - A$ is not positive definite on Range$(\widetilde{M}A)$.

Theorem 2.2 can be used to derive further conditions implying the convergence of iteration (1.4). The following result has the same spirit as Theorem 3.1, but note that the hypothesis (3.2) is not needed here. This result is used later in the paper.

THEOREM 3.4. [illegible] $A \in \mathbb{C}^{n\times n}$ [illegible] $H = I - \widetilde{M}A$ [illegible]

(3.4) $\widetilde{M} + \widetilde{M}^H - \widetilde{M}^H A\widetilde{M}$ [illegible] Range(A)

[illegible] $\|H\|_A < 1$ [illegible]. We have

$$H^H AH = A - A(\widetilde{M} + \widetilde{M}^H - \widetilde{M}^H A\widetilde{M})A. \tag{3.5}$$

So if $\widetilde{M} + \widetilde{M}^H - \widetilde{M}^H A\widetilde{M}$ is positive definite on Range(A), we immediately get that for $x \notin \text{Null}(A)$ we have $\|Hx\|_A < \|x\|_A$, i.e., $\|H\|_A < 1$. On the other hand, if $\|H\|_A < 1$ and $x \notin \text{Null}(A)$, then $\|Hx\|_A^2 < \|x\|_A^2$. From (3.5) we see that this means $\langle Ax, (\widetilde{M} + \widetilde{M}^H - \widetilde{M}^H A\widetilde{M})Ax\rangle > 0$, i.e., $(\widetilde{M} + \widetilde{M}^H - \widetilde{M}^H A\widetilde{M})$ is positive definite on Range(A). ∎

We note that for the matrices of Example 3.3 we have that

$$\widetilde{M} + \widetilde{M}^H - \widetilde{M}^H A\widetilde{M} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix},$$

which is the identity on Range(A), and thus positive definite on Range(A).

3.1. Application to additive Schwarz. We start this section with a general result where the generic operator $\widetilde{M}$ is decomposed into p operators and involves a positive damping factor θ; i.e., we have $\widetilde{M} = \theta \sum_{i=1}^{p} \widetilde{M}_i$ and

$$H = I - \widetilde{M}A = I - \theta \sum_{i=1}^{p} \widetilde{M}_i A. \tag{3.6}$$

As we shall see, this general formulation applies in particular to several variants of additive Schwarz iterations.

One of the hypotheses we use is that there exists a number $\gamma > 0$ such that

$$\Re\langle x, \widetilde{M}_i Ax\rangle_A \geq \gamma \cdot \langle \widetilde{M}_i Ax, \widetilde{M}_i Ax\rangle_A \text{ for all } x \in \mathbb{C}^n \text{ and for } i = 1, \ldots, p. \tag{3.7}$$

Here, $\Re z$ denotes the real part of a complex number z. It is easy to see that (3.7) is equivalent to the hypothesis (cf. (3.4))

$$\widetilde{M}_i + \widetilde{M}_i^H - 2\gamma \widetilde{M}_i^H A\widetilde{M}_i \text{ is positive semidefinite on Range}(A). \tag{3.8}$$

In the following theorems we give convergence results requiring upper bounds for the damping factor θ in (3.6). These upper bounds are given in terms of p (usually representing the number of subdomains). Nevertheless, in the same way as is done in the convergence analysis for classical additive Schwarz methods for Hermitian positive definite matrices, where q "colors" are used, the bounds can be enlarged to be in terms of q; see Remark 3.14.

THEOREM 3.5. [illegible] $A \in \mathbb{C}^{n\times n}$ [illegible] $\widetilde{M}_i \in \mathbb{C}^{n\times n}$, $i = 1, \ldots, p$ [illegible]

(i) [illegible] $\gamma > 0$ [illegible] (3.7) [illegible]

(ii) $\cap_{i=1}^{p}\mathrm{Null}(A\widetilde{M}_iA) = \mathrm{Null}(A)$

$\bar{\theta} \geq \frac{2\gamma}{p}$ $0 < \theta < \bar{\theta}$ H (3.6) $\|H\|_A < 1$ $\widetilde{M}$ (3.4)

(3.9) $|\langle \widetilde{M}_iAx, \widetilde{M}_jAx\rangle_A| \leq c_{ij} \cdot \|\widetilde{M}_iAx\|_A \cdot \|\widetilde{M}_jAx\|_A$ $x \in \mathbb{C}^n$, $i,j = 1,\ldots,p$,

$0 \leq c_{ij} = c_{ji} \leq 1$ $c_{ii} = 1$ $\bar{\theta}$ $\bar{\theta} \geq (2\gamma)/\lambda_{max}(C)$ $\lambda_{max}(C)$ $C = (c_{ij})$. For all $x \in \mathbb{C}^n$ we have

$$\langle Hx, Hx\rangle_A = \langle x,x\rangle_A - 2\theta\sum_{i=1}^{p}\Re\langle x, \widetilde{M}_iAx\rangle_A + \theta^2\sum_{i,j=1}^{p}\langle \widetilde{M}_iAx, \widetilde{M}_jAx\rangle_A. \tag{3.10}$$

For ease of notation we put

$$m_i = \|\widetilde{M}_iAx\|_A, \qquad i = 1,\ldots,p, \tag{3.11}$$

and observe that using hypothesis (i) it holds that

$$\gamma \cdot m_i^2 = \gamma \cdot \langle \widetilde{M}_iAx, \widetilde{M}_iAx\rangle_A \leq \Re\langle x, \widetilde{M}_iAx\rangle_A. \tag{3.12}$$

Also, using the Cauchy–Schwarz inequality, one has $\langle \widetilde{M}_iAx, \widetilde{M}_jAx\rangle_A \leq m_im_j$. Let now $m = (m_1,\ldots,m_p)^T$ and $E \in \mathbb{C}^{p\times p}$ be the matrix of all ones. Then from (3.10) we obtain

$$\begin{aligned}\langle Hx, Hx\rangle_A &\leq \langle x,x\rangle_A - 2\theta\gamma\sum_{i=1}^{p}m_i^2 + \theta^2\sum_{i,j=1}^{p}m_im_j \\ &= \langle x,x\rangle_A - \theta\cdot\langle m, (2\gamma I - \theta E)m\rangle. \end{aligned}\tag{3.13}$$

For $\theta < (2\gamma)/p$ the matrix $2\gamma I - \theta E$ is strictly diagonally dominant and thus Hermitian and positive definite. Therefore, once we have shown that $m \neq 0$ for $x \notin \mathrm{Null}(A)$ we will have proven the first part of the theorem, since then, by (3.13), we have $\langle Hx, Hx\rangle_A < \langle x,x\rangle_A$, i.e., $\|H\|_A < 1$. But if $m_i = 0$ for $i = 1,\ldots,p$, we have $\widetilde{M}_iAx \in \mathrm{Null}(A)$ and thus $x \in \mathrm{Null}(A\widetilde{M}_iA)$ for $i = 1,\ldots,p$. By (ii) this gives $x \in \mathrm{Null}(A)$.

The fact that $\widetilde{M}$ fulfills (3.4) follows directly from Theorem 3.4.

If the strengthened Cauchy–Schwarz inequalities (3.9) hold, we can replace (3.13) with the stronger

$$\langle Hx, Hx\rangle_A \leq \langle x,x\rangle_A - \theta\cdot\langle m, (2\gamma I - \theta C)m\rangle.$$

Since $2\gamma I - \theta C$ is Hermitian and positive definite for $\theta < (2\gamma)/\lambda_{\max}(C)$, the same arguments as before prove the last part of the theorem. ∎

THEOREM 3.6. $A \in \mathbb{C}^{n\times n}$ (i) (ii) 3.5 3.5 $q < p$ $i \in \{1,\ldots,p\}$ Range$(\widetilde{M}_i)$ Range$(\widetilde{M}_j), j = 1,\ldots,p$ $j \neq i$ $q-1$ $\bar{\theta}$ $\bar{\theta} \geq (2\gamma)/q$. By the hypothesis, we have strengthened Cauchy–Schwarz inequalities (3.9), where for each i at most q of the c_{ij} are nonzero, and the nonzero ones can be

chosen to be equal to 1. Therefore, all row sums of C are bounded by q, and thus by Gershgorin's theorem (see, e.g., [20]), we have $\lambda_{\max}(C) \le q$. ∎

In the results presented so far, the operators $\widetilde{M}_i$ were allowed to be of quite general nature; in particular, they may be non-Hermitian. In many situations, however, the operators $\widetilde{M}_i$ are Hermitian and positive semidefinite on Range(A). In this case, $x \in \text{Null}(A\widetilde{M}_i A)$ implies $0 = \langle x, (A\widetilde{M}_i A)x\rangle = \langle Ax, \widetilde{M}_i Ax\rangle$ and thus $Ax \in \text{Null}(\widetilde{M}_i)$, i.e., $x \in \text{Null}(\widetilde{M}_i A)$. In this situation we consequently have $\cap_{i=1}^p \text{Null}(A\widetilde{M}_i A) = \cap_{i=1}^p \text{Null}(\widetilde{M}_i A)$, which directly gives the following corollary to Theorem 3.5.

THEOREM 3.7. [illegible] $A \in \mathbb{C}^{n\times n}$ [illegible] $\widetilde{M}_i \in \mathbb{C}^{n\times n}$, $i = 1, \dots, p$, [illegible]

(i) $\widetilde{M}_i$ [illegible] $i = 1, \dots, p$

(ii) [illegible] $\gamma > 0$ [illegible] (3.7) [illegible]

(iii) $\cap_{i=1}^p \text{Null}(\widetilde{M}_i A) = \text{Null}(A)$

[illegible] 3.5 [illegible]

To study additive Schwarz methods, we need some further notation. We consider a decomposition of $\mathbb{C}^n$ into p subspaces of dimensions n_i, $i = 1, \dots, p$, represented by $\mathbb{C}^{n_i}$. By R_i we denote the projections ("restrictions") onto these subspaces, represented as matrices $R_i \in \mathbb{C}^{n_i \times n}$ having full rank n_i. We define the Galerkin operators

$$A_i = R_i A R_i^H \in \mathbb{C}^{n_i \times n_i}, \qquad i = 1, \dots, p.$$

The following result on the range of the Galerkin operator will be useful later.

LEMMA 3.8. [illegible] A [illegible]

$$\text{Range}(A_i) = \text{Range}(R_i A).$$

[illegible]. Since A_i is Hermitian, the assertion is equivalent to $\text{Null}(A_i) = \text{Null}(AR_i^H)$. Clearly, $\text{Null}(A_i) \supseteq \text{Null}(AR_i^H)$. On the other hand, if $x \in \text{Null}(A_i)$, it satisfies $0 = \langle A_i x, x\rangle = \langle AR_i^H x, R_i^H x\rangle$, which implies $R_i^H x \in \text{Null}(A)$, i.e., $x \in \text{Null}(AR_i^H)$, showing that we also have $\text{Null}(A_i) \subseteq \text{Null}(AR_i^H)$. ∎

For the moment, let us assume that, although A is only Hermitian positive semidefinite, all Galerkin operators are nonsingular (and thus positive definite); i.e., we assume that $\text{Range}(R_i^H) \cap \text{Null}(A) = \{0\}$ for all i; see, e.g., [4], [6], [15], [16], for examples when this situation occurs. The additive (damped) Schwarz iteration for solving $Ax = b$ is then given as (1.4) with

$$\widetilde{M} = \theta \sum_{i=1}^{p} R_i^H A_i^{-1} R_i \text{ and } H = I - \widetilde{M}A. \tag{3.14}$$

We refer the reader, e.g., to [17], [19], and references therein for details on Schwarz methods, and to [2], [9] for algebraic formulations.

THEOREM 3.9. [illegible] $A \in \mathbb{C}^{n\times n}$ [illegible] R_i [illegible]

$$\cap_{i=1}^p \text{Null}(R_i) = \{0\}, \tag{3.15}$$

[illegible] $A_i = R_i A R_i^H$ [illegible] $i = 1, \dots, p$ [illegible] $0 < \theta < \frac{2}{p}$ [illegible] H [illegible] (3.14) [illegible] $\|H\|_A < 1$ [illegible] $\widetilde{M}$ [illegible] (3.4)

[illegible]. We show that $\widetilde{M}_i = R_i^H A_i^{-1} R_i$ satisfies hypotheses (i)–(iii) of Theorem 3.7 with $\gamma = 1$. Obviously, $\widetilde{M}_i$ is Hermitian. For (ii) we have

$$\widetilde{M}_i A \widetilde{M}_i A = R_i^H A_i^{-1} R_i A R_i^H A_i^{-1} R_i A = R_i^H A_i^{-1} A_i A_i^{-1} R_i A = R_i^H A_i^{-1} R_i A = \widetilde{M}_i A,$$

which shows that the $\widetilde{M}_i A$ are projections, i.e., (ii) holds with $\gamma = 1$. For (iii) let $x \in \cap_{i=1}^{p} \mathrm{Null}(\widetilde{M}_i A)$; then

$$0 = \langle R_i^H A_i^{-1} R_i A x, x \rangle_A = \langle A_i^{-1} R_i A x, R_i A x \rangle,$$

which, since A_i is Hermitian positive definite, implies $R_i A x = 0$, $i = 1, \ldots, p$, i.e.,

$$A x \in \cap_{i=1}^{p} \mathrm{Null}(R_i) = \{0\}.$$

Thus, $x \in \mathrm{Null}(A)$. So we have $\cap_{i=1}^{p} \mathrm{Null}(\widetilde{M}_i A) \subseteq \mathrm{Null}(A)$, and since the opposite inclusion is trivial we have (iii). ∎

3.10. The restriction operators R_i in our formulation of Schwarz methods are very general. In the special case when they are Boolean gather operators (i.e., their rows being rows of the identity), using Theorem 3.9 we recover the convergence part of [16, Theorem 4.2].

Let us also note that for the "prolongation" operators $P_i = R_i^H$, we have $\mathrm{Range}(P_i) = \mathrm{Null}(R_i)^{\perp}$. Thus, condition (3.15) can equivalently be stated as

$$\sum_{i=1}^{p} \mathrm{Range}(P_i) = \mathbb{C}^n,$$

as is done, e.g., in [10].

Theorem 3.9 can be extended to the case where the Galerkin matrices A_i are singular, if we replace their inverses by the Moore-Penrose pseudoinverses $A_i^{\dagger}$.

THEOREM 3.11. [illegible] $A \in \mathbb{C}^{n \times n}$ [illegible] $0 < \theta < \frac{2}{p}$ [illegible]

$$H = I - \widetilde{M} A, \quad \widetilde{M} = \theta \sum_{i=1}^{p} R_i^H A_i^{\dagger} R_i, \quad A_i = R_i A R_i^H. \tag{3.16}$$

[illegible] R_i [illegible]

$$\cap_{i=1}^{p} \mathrm{Null}(R_i^H A_i^{\dagger} R_i A) = \mathrm{Null}(A).$$

[illegible] H [illegible] (3.16) [illegible] $\|H\|_A < 1$ [illegible] $\widetilde{M}$ [illegible] (3.4) [illegible]. All we need to do is show that with $\widetilde{M}_i = R_i^H A_i^{\dagger} R_i$, $i = 1, \ldots, p$, the hypotheses (i) and (ii) of Theorem 3.7 (with $\gamma = 1$) are satisfied, (iii) being assumed. Since A_i is Hermitian positive semidefinite, so is $A_i^{\dagger}$, and therefore also is $\widetilde{M}_i$. For (ii), we have

$$R_i^H A_i^{\dagger} R_i A R_i^H A_i^{\dagger} R_i A = R_i^H A_i^{\dagger} A_i A_i^{\dagger} R_i A = R_i^H A_i^{\dagger} R_i A \tag{3.17}$$

showing that the matrices $R_i^H A_i^{\dagger} R_i A$ are again projections, i.e., (ii) holds with equality. ∎

We next consider a situation usually referred to as inexact solution of the local problems; see, e.g., [1], [5], [17], [19]. This is the situation, e.g., when the solution of the local problem

$$A_i y_i = z_i \tag{3.18}$$

is not obtained exactly. Thus one replaces $A_i^{-1} z_i$ or $A_i^{\dagger} z_i$ with a vector other than a solution of (3.18), and this is represented by $\tilde{A}_i z_i$. In this case we have $\widetilde{M}_i = R_i^H \tilde{A}_i R_i$;

and using Lemma 3.8 it is easy to see that the hypothesis (3.8), or equivalently (3.7), can be rewritten as

$$\tilde{A}_i + \tilde{A}_i^H - 2\gamma \tilde{A}_i^H A_i \tilde{A}_i \text{ is positive semidefinite on Range}(A_i). \tag{3.19}$$

Observe that here $\tilde{A}_i$ is not assumed to be symmetric, and thus neither is $\widetilde{M}_i$.

We are ready now to establish the convergence of (damped) additive Schwarz iterations with inexact local solvers, which follows directly from Theorem 3.5.

THEOREM 3.12. [illegible] $A \in \mathbb{C}^{n\times n}$ [illegible] $A_i = R_i A R_i^H$, $i = 1, \ldots, p$ [illegible] $\tilde{A}_i$ [illegible] γ [illegible] (3.19) [illegible] $0 < \theta < (2\gamma)/p$ [illegible]

$$H = I - \widetilde{M}A, \text{ [illegible] } \widetilde{M} = \theta \sum_{i=1}^{p} \widetilde{M}_i = \theta \sum_{i=1}^{p} R_i^H \tilde{A}_i R_i.$$

[illegible] R_i [illegible]

$$\cap_{i=1}^{p} \text{Null}(A R_i^H \tilde{A}_i R_i A) = \text{Null}(A).$$

[illegible] H [illegible] $\|H\|_A < 1$ [illegible] $\widetilde{M}$ [illegible] (3.4)

Note that condition (3.19) is fulfilled with $\gamma = \frac{1}{2}$ if $\tilde{A}_i + \tilde{A}_i^H - \tilde{A}_i^H A_i \tilde{A}_i$ is positive definite on Range(A_i), which is precisely (3.4) from Theorem 3.4. So in this special case, by Theorem 3.4, we have $\|I - \tilde{A}_i A_i\|_{A_i} < 1$, or that an iteration for the solution of the local problem (3.18) with iteration matrix $I - \tilde{A}_i A_i$ is convergent. One particular general example of this situation is when one uses a splitting of $A_i = B_i - C_i$, and the solution of the system (3.18) is approximated by κ classical stationary iterations associated with this splitting. Thus, for this example

$$\tilde{A}_i = \sum_{j=0}^{\kappa-1} (B_i^{-1} C_i)^j B_i^{-1}. \tag{3.20}$$

Of course one can have different values of κ for different local problems. As a particular case, consider the canonical decompositions $A_i = D_i - L_i - L_i^H$ and put $B_i = \frac{1}{\omega} D_i - L_i$, i.e., relaxed Gauss–Seidel. If one sets $\tilde{A}_i = B_i^{-1}$ the local solutions are approximated by one step of the relaxed Gauss–Seidel, i.e., $\kappa = 1$. Assuming that no diagonal element of A_i is zero and that $\omega \in (0, 2)$, a simple calculation shows that (3.19) is fulfilled with $\gamma = \frac{1}{2}$. Since we then have that the relaxed Gauss–Seidel iteration is convergent, using Theorem 3.4, we see that the $\tilde{A}_i$ of (3.20) also fulfills (3.19) with $\gamma = \frac{1}{2}$ for all integer values of κ.

[illegible] 3.13. We note that a special case of Theorem 3.12 when R_i are Boolean gather operators and $\tilde{A}_i$ are symmetric and nonsingular is [16, Theorem 6.1], where the hypothesis used there is equivalent to

$$\langle z, \tilde{A}_i z \rangle \le \langle z, A_i^{-1} z \rangle \quad \text{for all } z \in \mathbb{C}^{n_i} \text{ and for } i = 1, \ldots, p,$$

which implies (3.19) with $\gamma \le 1$. Indeed, in this case, we have that the difference $\tilde{A}_i^{-1} - A_i$ is positive semidefinite, and we write

$$2\tilde{A}_i - 2\gamma \tilde{A}_i^H A_i \tilde{A}_i = 2\tilde{A}_i \left(\tilde{A}_i^{-1} - \gamma A_i \right) \tilde{A}_i.$$

3.14. If in Theorems 3.7, 3.9, 3.11, and 3.12, we add the hypothesis that there exists a natural number $q < p$ such that for each $i \in \{1, \ldots, p\}$ the space $\text{Range}(R_i^H)$ is orthogonal to all spaces $\text{Range}(R_j^H)$, $j = 1, \ldots, p$, $j \neq i$, except for at most $q - 1$ such indices, then, using Theorem 3.6, the results hold for $\theta < 2/q$ (or $\theta < (2\gamma)/q$ in Theorem 3.12); cf. [10, Chapter 11.2.4], where this is done for classical additive Schwarz for A Hermitian positive definite. See also [2], [9], and [16] for other such situations.

We note that Hermitian positive semidefinite matrices $\widetilde{M}_i$ different from those considered in Theorems 3.9, 3.11, and 3.12 also do appear in other Schwarz contexts, and our general Theorem 3.5 would apply to such cases as well. For example, in [14] matrices of the form $\widetilde{M}_i = R_i^H(A_i + G_i)R_i$ are used, where G_i derives from the Robin boundary conditions.

3.2. Multiplicative Schwarz. Instead of an additive we now consider a multiplicative combination of p operators $\widetilde{M}_i$ resulting in

$$H = (I - \widetilde{M}_p A)(I - \widetilde{M}_{p-1}A) \cdots (I - \widetilde{M}_1 A) = \prod_{i=p}^{1} (I - \widetilde{M}_i A). \tag{3.21}$$

Of course, the iteration operator H can be written in the form $H = I - \widetilde{M}A$, but an explicit formula for $\widetilde{M}$ is not needed in our convergence analysis. As in the additive case, the general formulation (3.21) applies, for particular choices of the matrices $\widetilde{M}_i$, to several variants of multiplicative Schwarz methods, including, for example, those corresponding to Robin boundary conditions [14].

As we did in the additive case, we first state a general theorem which we then apply to the multiplicative Schwarz setting.

THEOREM 3.15. [illegible] $A \in \mathbb{C}^{n\times n}$ [illegible] $\widetilde{M}_i \in \mathbb{C}^{n\times n}$, $i = 1, \ldots, p$, [illegible]

(i) A [illegible] $\widetilde{M}_i A$, $i = 1, \ldots, p$, [illegible] $\text{Null}(A\widetilde{M}_i A) = \text{Null}(\widetilde{M}_i A)$, $i = 1, \ldots, p$

(ii) [illegible] $\gamma > \frac{1}{2}$ [illegible] (3.7) [illegible]

(iii) $\cap_{i=1}^{p} \text{Null}(\widetilde{M}_i A) = \text{Null}(A)$

[illegible] H [illegible] (3.21) [illegible] H [illegible] $\|H\|_A < 1$ [illegible] $H = I - \widetilde{M}A$ [illegible] $\widetilde{M}$ [illegible] (3.4)

[illegible]. We first note that by (ii) we have, for $x \in \mathbb{C}^n$ and $i = 1, \ldots, p$,

$$\begin{aligned} \langle (I - \widetilde{M}_i A)x, (I - \widetilde{M}_i A)x\rangle_A &= \langle x, x\rangle_A - 2\Re\langle x, \widetilde{M}_i Ax\rangle_A + \langle \widetilde{M}_i Ax, \widetilde{M}_i Ax\rangle_A \\ &\leq \langle x, x\rangle_A - (2\gamma - 1)\langle \widetilde{M}_i Ax, \widetilde{M}_i Ax\rangle_A. \end{aligned} \tag{3.22}$$

Now let $x^{(1)} = z$ and $x^{(i+1)} = (I - \widetilde{M}_i A)x^{(i)}$, $i = 1, \ldots, p$, so that $x^{(p+1)} = Hx^{(1)} = Hz$. Using (3.22) repeatedly we obtain

$$\langle Hz, Hz\rangle_A - \langle z, z\rangle_A \leq -(2\gamma - 1)\sum_{i=1}^{p} \langle \widetilde{M}_i Ax^{(i)}, \widetilde{M}_i Ax^{(i)}\rangle_A. \tag{3.23}$$

The right-hand side of (3.23) is nonpositive. It remains to show that it is zero only when $z \in \text{Null}(A)$. Now the right-hand side of (3.23) is zero if and only if $\langle \widetilde{M}_i Ax^{(i)}, \widetilde{M}_i Ax^{(i)}\rangle_A = 0$ for $i = 1, \ldots, p$. This is equivalent to $\widetilde{M}_i Ax^{(i)} \in \text{Null}(A)$, i.e., $x^{(i)} \in \text{Null}(A\widetilde{M}_i A)$, which by assumption (i) implies $x^{(i)} \in \text{Null}(\widetilde{M}_i A)$ for

$i = 1, \ldots, p$. But then $x^{(i+1)} = (I - \widetilde{M}_i A)x^{(i)} = x^{(i)}$ for $i = 1, \ldots, p$, resulting in $x^{(i)} = z$ for $i = 1, \ldots, p$, and $z \in \mathrm{Null}(\widetilde{M}_i A)$ for $i = 1, \ldots, p$. By assumption (iii) this means $z \in \mathrm{Null}(A)$. So we have shown $\|H\|_A < 1$. The fact that $\widetilde{M}$ fulfills condition (3.4) follows directly from Theorem 3.4. ∎

We now use Theorem 3.15 for the analysis of multiplicative Schwarz methods. We use the notation introduced in section 3.1. As in the additive case, we first consider the case where the Galerkin operators $A_i = R_i A R_i^H$ are nonsingular, i.e., we have

$$\widetilde{M}_i = R_i^H A_i^{-1} R_i, \qquad i = 1, \ldots, p. \tag{3.24}$$

THEOREM 3.16. [illegible] $A \in \mathbb{C}^{n \times n}$ [illegible] R_i [illegible]

$$\cap_{i=1}^{p} \mathrm{Null}(R_i) = \{0\},$$

[illegible] $A_i = R_i A R_i^H$ [illegible] $i = 1, \ldots, p$ [illegible] H [illegible] (3.21) [illegible] $\widetilde{M}_i$ [illegible] (3.24) [illegible] $\|H\|_A < 1$ [illegible] $H = I - \widetilde{M}A$ [illegible] $\widetilde{M}$ [illegible] (3.4)

[illegible]. We need to show that the hypotheses (i)–(iii) of Theorem 3.15 are fulfilled with $\gamma = 1$. For (ii) and (iii), this was already done in the proof of Theorem 3.9. To show that (i) holds, we first note that, trivially, $\mathrm{Null}(A\widetilde{M}_i A) \supseteq \mathrm{Null}(\widetilde{M}_i A)$. On the other hand, $x \in \mathrm{Null}(A\widetilde{M}_i A)$ implies $0 = \langle x, A\widetilde{M}_i Ax \rangle = \langle Ax, \widetilde{M}_i Ax \rangle$, which, since $\widetilde{M}_i$ is Hermitian positive semidefinite, yields $Ax \in \mathrm{Null}(\widetilde{M}_i)$, i.e., $x \in \mathrm{Null}(\widetilde{M}_i A)$. ∎

The next theorem considers the case where the Galerkin operators can be singular.

THEOREM 3.17. [illegible] $A \in \mathbb{C}^{n \times n}$ [illegible] $A_i = R_i A R_i^H$ [illegible] $\widetilde{M}_i = R_i^H A_i^{\dagger} R_i$ [illegible] $i = 1, \ldots, p$ [illegible] H [illegible] (3.21) [illegible] R_i [illegible]

$$\cap_{i=1}^{p} \mathrm{Null}(R_i^H A_i^{\dagger} R_i A) = \mathrm{Null}(A).$$

[illegible] H [illegible] $\|H\|_A < 1$ [illegible] $H = I - \widetilde{M}A$ [illegible] $\widetilde{M}$ [illegible] (3.4)

[illegible]. The proof follows again by showing that assumptions (i) to (iii) of Theorem 3.15 are fulfilled with $\gamma = 1$. But (iii) is assumed and (i) follows in exactly the same manner as in the proof of Theorem 3.16, whereas (ii) holds with $\gamma = 1$ since the $\widetilde{M}_i A$ are projections as shown in (3.17). ∎

We end this section considering multiplicative Schwarz iterations with inexact solutions of the local problems (3.18), i.e., when $\widetilde{M}_i = R_i^H \tilde{A}_i R_i$.

THEOREM 3.18. [illegible] $A \in \mathbb{C}^{n \times n}$ [illegible] $A_i = R_i A R_i^H$, $i = 1, \ldots, p$ [illegible] $\widetilde{M}_i = R_i^H \tilde{A}_i R_i$ [illegible] $\tilde{A}_i$ [illegible]

(a) $\tilde{A}_i$ [illegible]

(b) $\tilde{A}_i + \tilde{A}_i^H$ [illegible] $\mathrm{Range}(A_i)$

[illegible] $\gamma > \frac{1}{2}$ [illegible] (3.19) [illegible] R_i [illegible]

$$\cap_{i=1}^{p} \mathrm{Null}(R_i^H \tilde{A}_i R_i A) = \mathrm{Null}(A).$$

[illegible] H [illegible] (3.21) [illegible] $\|H\|_A < 1$ [illegible] $H = I - \widetilde{M}A$ [illegible] $\widetilde{M}$ [illegible] (3.4)

. We have to prove that assumptions (i) and (ii) of Theorem 3.15 hold, (iii) being part of the assumptions. Recall that (i) from Theorem 3.15 says that $\mathrm{Null}(A\widetilde{M}_i A) = \mathrm{Null}(\widetilde{M}_i A)$, where only the inclusion $\mathrm{Null}(A\widetilde{M}_i A) \subseteq \mathrm{Null}(\widetilde{M}_i A)$ is nontrivial. In the case that $\tilde{A}_i$ is Hermitian positive definite, $\widetilde{M}_i$ is Hermitian positive definite, too, and (i) of Theorem 3.15 follows as in the proof of Theorem 3.16. In the case that $\tilde{A}_i + \tilde{A}_i^H$ is positive definite on $\mathrm{Range}(A_i)$, assume that $A\widetilde{M}_i Ax = 0$. Then

$$\begin{aligned} 0 = \langle x, AR_i^H \tilde{A}_i R_i Ax\rangle &= \langle R_i Ax, \tilde{A}_i R_i Ax\rangle, \\ 0 = \langle AR_i^H \tilde{A}_i R_i Ax, x\rangle &= \langle R_i Ax, \tilde{A}_i^H R_i Ax\rangle, \end{aligned}$$

and thus $0 = \langle R_i Ax, (\tilde{A}_i + \tilde{A}_i^H) R_i Ax\rangle$. By Lemma 3.8, we have $R_i Ax \in \mathrm{Range}(A_i)$, and since $\tilde{A}_i + \tilde{A}_i^H$ is positive definite on that space we get $R_i Ax = 0$. This yields $R_i^H \tilde{A}_i R_i Ax = 0$, i.e., $x \in \mathrm{Null}(\widetilde{M}_i A)$, so that we have again shown that (i) of Theorem 3.15 holds.

Finally, since (3.19) is equivalent to (3.7), we also have (ii). ∎

We observe again that assuming $\|I - \tilde{A}_i A_i\|_{A_i} < 1$ is sufficient for (3.19) to hold. Indeed, by Theorem 3.4, this assumption is equivalent to that $\tilde{A}_i + \tilde{A}_i^H - \tilde{A}_i^H A_i \tilde{A}_i$ is positive definite on $\mathrm{Range}(A_i)$, which implies that $\tilde{A}_i + \tilde{A}_i^H$ is positive definite on $\mathrm{Range}(A_i)$ and that $\tilde{A}_i + \tilde{A}_i^H - 2\gamma \tilde{A}_i^H A_i \tilde{A}_i$ is still positive semidefinite on $\mathrm{Range}(A_i)$ for $\gamma > \frac{1}{2}$ sufficiently close to $\frac{1}{2}$. Hence, we have the following corollary.

COROLLARY 3.19. . $A \in \mathbb{C}^{n\times n}$, . $A_i = R_i A R_i^H$, $i = 1, \ldots, p$. $\widetilde{M}_i = R_i^H \tilde{A}_i R_i$ . . $\tilde{A}_i$.

$$\|I - \tilde{A}_i A_i\|_{A_i} < 1. \tag{3.25}$$

. R_i .

$$\cap_{i=1}^{p} \mathrm{Null}(R_i^H \tilde{A}_i R_i A) = \mathrm{Null}(A).$$

. H . (3.21) . $\|H\|_A < 1$ . . $H = I - \widetilde{M}A$. $\widetilde{M}$. (3.4)

Again, one can use relaxed Gauss–Seidel, i.e., $\tilde{A}_i = B_i = \frac{1}{\omega} D_i - L_i$, where $A_i = D_i - L_i - L_i^H$. We have $\tilde{A}_i + \tilde{A}_i^H = \frac{2-\omega}{\omega} D + A$, which is positive definite for $\omega \in (0, 2)$ if A_i has no zero diagonal elements. Thus, assumption (ii) of Theorem 3.18 and assumption (3.25) of Corollary 3.19 are fulfilled in this case, as well as for $\tilde{A}_i$ as in (3.20).

We note also that for $\tilde{A}_i$ nonsingular, [16, Theorem 6.4] follows from Theorem 3.18, since in [16] it is assumed that $\tilde{A}_i^{-1} + \tilde{A}_i^{-H} - A_i$ is positive definite. This assumption can be written $\tilde{A}_i^{-H}(\tilde{A}_i^H + \tilde{A}_i - \tilde{A}_i^H A_i \tilde{A}_i)\tilde{A}_i^{-1}$ being positive definite, so that (3.19) holds for some $\gamma > 1/2$.

4. Conclusions. We presented a very general convergence result for stationary iterative methods for linear systems whose coefficient matrix A is Hermitian and positive semidefinite. It is shown that if for $x \notin \mathrm{Null}(A)$, $\langle x, Hx\rangle_A < \langle x, x\rangle_A$, with $H = I - \widetilde{M}A$, then $\widetilde{M}$ is injective on $\mathrm{Range}(A)$, and H is semiconvergent. This result allowed us to give simple proofs of well-known results as well as to generalize them in several directions. We further used these new results to give convergence proofs of several variants of additive and multiplicative Schwarz iterations. These variants include those with local problems with Neumann or Robin boundary conditions as well as the inexact solution of the local problems.

Acknowledgments. We thank Zhi-Hao Cao and Yimin Wei for making available to us advanced copies of their papers [7], [13]. We also thank them, Sébastien Loisel, and two anonymous referees for their comments on an earlier version of the paper.

REFERENCES

[1] J. Arnal, V. Migallón, J. Penadés, and D. B. Szyld, *Newton additive and multiplicative Schwarz iterative methods*, IMA J. Numer. Anal., 28 (2008), pp. 143–161.

[2] M. Benzi, A. Frommer, R. Nabben, and D. B. Szyld, *Algebraic theory of multiplicative Schwarz methods*, Numer. Math., 89 (2001), pp. 605–639.

[3] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.

[4] S. Borovac, *A Graph Based Approach to the Convergence of One Level Schwarz Iterations for Singular M-Matrices and Markov Chains*, Technical Report BUW-SC 2007/3, Applied Computer Science Group, University of Wuppertal, Germany, 2007.

[5] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, *Analysis of non-overlapping domain decomposition algorithms with inexact solves*, Math. Comp., 67 (1998), pp. 1–19.

[6] R. Bru, F. Pedroche, and D. B. Szyld, *Additive Schwarz iterations for Markov chains*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 445–458.

[7] Z.-H. Cao, *On the convergence of general stationary linear iterative methods for singular linear systems*, SIAM J. Matrix Anal. Appl., 29 (2008), pp. 1382–1388.

[8] S. Friedhoff, A. Frommer, and M. Heming, *Algebraic Multigrid Methods for Laplacians of Graphs*, in preparation.

[9] A. Frommer and D. B. Szyld, *Weighted max norms, splittings, and overlapping additive Schwarz iterations*, Numer. Math., 83 (1999), pp. 259–278.

[10] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, Springer, New York, Berlin, Heidelberg, 1994.

[11] H. B. Keller, *On the solution of singular and semidefinite linear systems by iteration*, SIAM J. Numer. Anal., 2 (1965), pp. 281–290.

[12] Y.-J. Lee, J. Wu, J. Xu, and L. Zikatanov, *On the convergence of iterative methods for semidefinite linear systems*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 634–641.

[13] L. Lin, Y. Wei, C.-W. Woo, and J. Zhou, *On the convergence of splittings for semidefinite linear systems*, Linear Algebra Appl., (2008), doi:10.1016/j.laa.2007.12.019.

[14] S. Loisel and D. B. Szyld, *On the Convergence of Algebraic Optimizable Schwarz Methods with Applications to Elliptic Problems*, Research Report 07-11-16, Department of Mathematics, Temple University, Philadelphia, PA, November 2007.

[15] I. Marek and D. B. Szyld, *Algebraic Schwarz methods for the numerical solution of Markov chains*, Linear Algebra Appl., 386 (2004), pp. 67–81.

[16] R. Nabben and D. B. Szyld, *Schwarz iterations for symmetric positive semidefinite problems*, SIAM J. Matrix Anal. Appl., 29 (2006), pp. 98–116.

[17] B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, New York, Melbourne, 1996.

[18] D. B. Szyld, *Equivalence of convergence conditions for iterative methods for singular equations*, Numer. Linear Algebra Appl., 1 (1994), pp. 151–154.

[19] A. Toselli and O. Widlund, *Domain Decomposition Methods—Algorithms and Theory*, Springer Ser. Comput. Math. 34, Springer, Berlin, Heidelberg, New York, 2005.

[20] R. S. Varga, *Matrix Iterative Analysis*, 2nd ed., revised and expanded, Springer, Berlin, Heidelberg, New York, 2000.

SPECIAL ISSUE ON TENSOR DECOMPOSITIONS AND APPLICATIONS

This issue of [illegible] was motivated by the Workshop on Tensor Decompositions and Applications, held in Luminy, France from August 29 to September 2, 2005. The issue was announced through both the SIMAX and the workshop web sites.

Though decompositions of higher-order tensors have been around for many years, the door is now opening on greater mathematical understanding and new applications. Tensor decompositions have been applied by researchers in psychometrics and chemometrics since the seventies. More recently, tensors have found their way to signal processing via the use of high-order statistics, joint matrix techniques, and applications in telecommunications. Other applications include complexity theory, (blind) system identification, biomedical engineering, numerical analysis, and data mining, among others.

This volume contains 15 papers that together form a nice cross-section of current research on tensor decompositions. The papers present new algorithms, basic algebraic results including the introduction of new decompositions, and applications in signal processing, scientific computing, and large-scale problems.

Many thanks are due to all the authors for their valuable contributions. We would also like to thank Henk van der Vorst, Mitch Chernoff, and the SIAM staff for their support.

Since the initial submission of the papers, two of the authors have passed away. Richard Harshman and Gene Golub were both authorities in their field.

Lieven De Lathauwer
Katholieke Universiteit Leuven

Pierre Comon
Centre National de la Recherche Scientifique

Nicola Mastronardi
Istituto per le Applicazioni del Calcolo “M. Picone”

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 939–956

TUCKER DIMENSIONALITY REDUCTION OF THREE-DIMENSIONAL ARRAYS IN LINEAR TIME*

I. V. OSELEDETS†, D. V. SAVOSTIANOV†, AND E. E. TYRTYSHNIKOV†

Abstract. We consider Tucker-like approximations with an $r \times r \times r$ core tensor for three-dimensional $n \times n \times n$ arrays in the case of $r \ll n$ and possibly very large n (up to 10^4–10^6). As the approximation contains only $\mathcal{O}(rn + r^3)$ parameters, it is natural to ask if it can be computed using only a small amount of entries of the given array. A similar question for matrices (two-dimensional tensors) was asked and positively answered in [S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *A theory of pseudo-skeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21]. In the present paper we extend the positive answer to the case of three-dimensional tensors. More specifically, it is shown that if the tensor admits a good Tucker approximation for some (small) rank r, then this approximation can be computed using only $\mathcal{O}(nr)$ entries with $\mathcal{O}(nr^3)$ complexity.

Key words. multidimensional arrays, Tucker decomposition, tensor approximations, low-rank approximations, skeleton decompositions, dimensionality reduction, data compression, large-scale matrices, data-sparse methods

AMS subject classifications. 15A69, 15A18, 65F30

DOI. 10.1137/060655894

1. Introduction.

Multidimensional arrays of data appear in many different applications. One can mention signal processing, statistics [3, 1, 4], chemometrics [5], face recognition [7], and solving multidimensional integral and differential equations [6] (a very comprehensive list of references on the subject can be found on the Three-Mode Company's Web site, [2]). These arrays often cannot be handled by standard methods because of their huge sizes: we cannot solve linear systems or calculate required decompositions due to speed or memory restrictions. The obvious solution is to perform a sort of [illegible] : an initial "large" array is transformed to a smaller array for which we can use standard methods. However, such a reduction by conventional approaches may be computationally still too expensive. In this paper we suggest a way to make it not only feasible but even quite fast. We will focus only on three-dimensional arrays mostly to simplify the presentation and note that our results can be generalized to more dimensions.

The most useful method to reduce dimension is based on the celebrated [illegible] [22] and solves the following problem: [illegible] ([illegible]) $\mathcal{A} = [a_{ijk}]$, $i = 1, \ldots, n_1$, $j = 1, \ldots, n_2$, $k = 1, \ldots, n_3$, [illegible]

[illegible]

$$a_{ijk} = \sum_{i'=1}^{r_1} \sum_{j'=1}^{r_2} \sum_{k'=1}^{r_3} g_{i'j'k'} u_{ii'} v_{jj'} w_{kk'} + e_{ijk} \tag{1.1}$$

[illegible] e_{ijk} [illegible] r_1, r_2, r_3. The matrices $U = [u_{ii'}]$,

*Received by the editors March 31, 2006; accepted for publication (in revised form) by N. Mastronardi October 11, 2006; published electronically September 25, 2008. This research was supported by the Russian Fund of Basic Research (grants 05-01-00721, 04-07-90336, and 06-01-08052) and a Priority Research Grant ONM-3 from the Department of Mathematical Sciences of the Russian Academy of Sciences.

http://www.siam.org/journals/simax/30-3/65589.html

†Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Street, 8 IVM RAN, Moscow 119991, Russia (ivan@bach.inm.ras.ru, draug@bach.inm.ras.ru, tee@bach.inm.ras.ru).

$V = [v_{jj'}]$, $W = [w_{kk'}]$ will be referred to as , and the $r_1 \times r_2 \times r_3$ tensor $\mathcal{G} = [g_{i'j'k'}]$ as the .

A well-known method for the computation of the Tucker decomposition is based on the SVD algorithm. Consider three rectangular "unfolding" matrices of appropriate sizes $A^{(1)}, A^{(2)}, A^{(3)}$, which contain n-mode vectors (columns, rows, and fibers, respectively) of the tensor $\mathcal{A}$. The left ("short") singular vectors of the SVDs of these matrices

$$A^{(1)} = U\Sigma_1\Phi_1^\top, \qquad A^{(2)} = V\Sigma_2\Phi_2^\top, \qquad A^{(3)} = W\Sigma_3\Phi_3^\top \tag{1.2}$$

give the factors U, V, W of the Tucker decomposition, possibly after an appropriate truncation, and the core is computed as

$$g_{i'j'k'} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk}u_{ii'}v_{jj'}w_{kk'}. \tag{1.3}$$

The tensor dimension can be large (for example, $n = 10^4 - 10^6$ for some tensors coming from three-dimensional integral equations). The array itself cannot even be stored in the operative memory as $\mathcal{O}(n^3)$ memory cells are needed. The computation of the SVDs in (1.2) by standard methods costs $\mathcal{O}(n^4)$ operations and is prohibitive for $n \geq 1000$.

However, we are chiefly interested in the case $r \ll n$, and the Tucker decomposition contains only $\mathcal{O}(rn + r^3)$ parameters. If a good approximation exists, we can ask if it can be computed using only a small amount of entries of the tensor $\mathcal{A}$. A similar question for matrices (two-dimensional tensors) was asked and positively answered in [14]. In the present paper we extend the positive answer to the case of three-dimensional tensors. More specifically, it will be shown that if the tensor admits a good Tucker approximation for some (small) rank r, then this approximation can be computed using only $\mathcal{O}(nr)$ entries with $\mathcal{O}(nr^3)$ complexity.

Prior to investigation of special low-parametric (data-sparse) representations obtained only from the knowledge of a small portion of the data entries, we use a general assumption that low-parametric approximations exist. In other words, we consider the cases with sufficiently small approximate tensor rank estimates. Several estimates for many interesting for practial purposes cases are developed in [15, 18, 19]. We can mention also some practical algorithms using interpolation and other function approximation techniques or additional structural properties rather than the given arrays of data [15, 21]. [20] is closest to the paradigm of a completely data-based method (using no knowledge beyond the data themselves); however, [20] contains no proof for the existence of a sufficiently good low-rank representation and does not suggest a general adaptive procedure for selecting "most meaningful" entries. Recently, much attention has been paid to the approximation of a given matrix by a low-rank matrix using randomized algorithms, for example, see [23]. To our best knowledge, these algorithms are fast only asymptotically with very large constants in the estimates and cannot be applied in practice. Moreover, the authors do not report any numerical results in their articles, so we cannot compare their methods with our method. In this paper we present the existence results and the adaptive three-dimensional cross algorithms.

2. Notations and definitions. Let us recall some basic facts about tensors [10, 11].

DEFINITION 2.1. $n_1 \times n_2 \times n_3$ $\mathcal{A}$ $A^{(1)}$ $n_1 \times n_2 n_3$, $A^{(2)}$ $n_2 \times n_1 n_3$, $A^{(3)}$ $n_3 \times n_1 n_2$

$$A^{(1)} = [a^1_{i(jk)}] = [a_{ijk}], \quad A^{(2)} = [a^2_{j(ki)}] = [a_{ijk}], \quad A^{(3)} = [a^3_{k(ij)}] = [a_{ijk}]. \tag{2.1}$$

$1, 2, 3$ ( ) ”

DEFINITION 2.2. $n_1 \times n_2 \times n_3$ $\mathcal{A} = [a_{ijk}]$

$$||\mathcal{A}|| = ||\mathcal{A}||_F = \left(\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} a^2_{ijk} \right)^{1/2}.$$

$$||\mathcal{A}||_C = \max_{i,j,k} |a_{ijk}|$$

Chebyshev norm $\mathcal{A}$.

DEFINITION 2.3 (outer product). $\mathcal{A}$ $a_{i_1,i_2,\ldots,i_p}$ $\mathcal{B}$ $b_{j_1,j_2,\ldots,j_q}$, $\mathcal{C} = \mathcal{A} \otimes B$ (+)

$$c_{i_1,i_2,\ldots,i_p,j_1,j_2,\ldots,j_q} = a_{i_1,i_2,\ldots,i_p} b_{j_1,j_2,\ldots,j_q}.$$

Tensors can be multiplied by matrices along a specified index (mode) direction.

DEFINITION 2.4 (mode convolution or n-mode product). $\mathcal{A} = [a_{ijk}]$ $n_1 \times n_2 \times n_3$ U $m_1 \times n_1$ $\mathcal{B} = [b_{ijk}]$ $m_1 \times n_2 \times n_3$

$$\mathcal{B} = \mathcal{A} \times_i U, \qquad b_{ijk} = \sum_{i'=1}^{n_1} u_{ii'} a_{i'jk}.$$

The operations $\mathcal{A} \times_j U$, $\mathcal{A} \times_k U$ are defined analogously, provided that U and $\mathcal{A}$ have appropriate sizes. In this notation, the Tucker decomposition (1.1) can be written as

$$\mathcal{A} = \mathcal{G} \times_i U \times_j V \times_k W.$$

We will say that a tensor has a (r_1, r_2, r_3) ( ) if (1.1) holds [10, 11].

The important objects are of the three-dimensional arrays.

DEFINITION 2.5. $\mathcal{A} = [a_{ijk}]$ $n_1 \times n_2 \times n_3$ k [1] $n_1 \times n_2$ A_k

$$(A_k)_{ij} = a_{ijk}.$$

The “short” vectors along the modes i, j, and k will be referred to as and , respectively.

[1] The slices by other two indices can also be defined, but it may lead to ambiguity. Since in this paper only kth slices are used, we omit these definitions here.

3. Existence theory. Suppose an $n_1 \times n_2 \times n_3$ tensor $\mathcal{A} = [a_{ijk}]$ is given and there exists a rank-(r_1, r_2, r_3) Tucker approximation to $\mathcal{A}$ with the accuracy ε:

$$\mathcal{A} = \mathcal{G} \times_i U \times_j V \times_k W + \mathcal{E}, \quad ||\mathcal{E}|| = \varepsilon. \tag{3.1}$$

If we are aware that such an approximation exists, then a generally different approximation of the same type with the accuracy bound $c\varepsilon$ (where $c > 1$ is a [illegible]) can be constructed from the knowledge of roughly the same amount of entries as those explicitly involved in (3.1). We want to prove this together with a bound on the deterioration coefficient c depending only upon dimensions and ranks but not on the entries of the array.

THEOREM 3.1. [illegible] (3.1) [illegible] $U, V,$ [illegible] W [illegible] $\mathcal{G}$. [illegible] $U', V',$ [illegible] W' [illegible] $n_1 \times r_1$, $n_2 \times r_2$, [illegible] $n_3 \times r_3$ [illegible] r_1 [illegible] r_2 [illegible] r_3 [illegible] $\mathcal{A}$, [illegible] $\mathcal{G}'$ [illegible]

$$\mathcal{A} = \mathcal{G}' \times_i U' \times_j V' \times_k W' + \mathcal{E}',$$

[illegible]

$$||\mathcal{E}'||_C \le (r_1 r_2 r_3 + 2r_1 r_2 + 2r_1 + 1)\varepsilon.$$

[illegible] Consider an unfolding matrix $A^{(1)}$ of the array $\mathcal{A}$. Since $\mathcal{A}$ has a rank-(r_1, r_2, r_3) approximation with accuracy ε, it is easy to see from (3.1) that $A^{(1)}$ has a rank-r_1 approximation with the same accuracy. A low-rank matrix can be approximated by its [illegible]

$$A^{(1)} = C\hat{C}^{-1}B^\top + \mathcal{E}_1,$$

where C is a $n_1 \times r_1$ matrix containing some r_1 columns of $A^{(1)}$, B is a $n_2 n_3 \times r_1$ matrix containing some r_1 rows of $A^{(1)}$, and $\hat{C}$ is a submatrix on the intersection of these rows and columns. In [13] it was proved that if $\hat{C}$ is a [illegible] (that is, the $r_1 \times r_1$ submatrix which has the largest absolute value of the determinant) in $\mathcal{A}^{(1)}$, then ε_1 is bounded as follows:

$$||\mathcal{E}_1||_C \le (r_1 + 1)\varepsilon,$$

where $||\cdot||_C$ denotes the largest magnitude element of a matrix (array). Also, if $\hat{C}$ is a maximal volume submatrix, it is easy to prove (cf. [13]) that the elements of $C\hat{C}^{-1}$ are not greater than 1 in modulus. Consequently,

$$\left| a_{ijk} - \sum_{s=1}^{r_1} \gamma_{is} z_{jks} \right| \le (r_1 + 1)\varepsilon,$$

where

$$|\gamma_{is}| \le 1, \qquad 1 \le i \le n_1,$$

and z_{jks} are in a one-to-one correspondence with the entries of $B^\top$ (for a fixed s these elements present a slice by the index i of the array $\mathcal{A}$). Also note that

$$\gamma_{is} = \sum_{l=1}^{r_1} u_{il}\phi_{ls},$$

where the matrix $[u_{il}]$ consists of some columns of $\mathcal{A}$.

Now let us look more closely at the matrix $B^\top$. In a reshaped form, it becomes the tensor with the elements z_{jks}. As previously, unfold this tensor along the index j. The ε-rank[2] of the unfolding matrix does not exceed r_2.

Again using the result of [13], we obtain the following inequalities:

$$\left| z_{jks} - \sum_{t=1}^{r_2} \sum_{\tau=1}^{r_2} \psi_{t\tau} v_{jt} w_{ks\tau} \right| \le (r_2 + 1)\varepsilon,$$

where the arrays $[v_{jt}]$ and $[w_{ks\tau}]$ consist of some rows and fibers of $\mathcal{A}$.

Unfolding the array $[w_{ks\tau}]$ by the index k, we observe that the ε-rank of the unfolding matrix cannot be larger than k_3. Hence, this matrix admits the skeleton approximation with the error bound

$$\left| w_{ks\tau} - \sum_{\alpha=1}^{k_3} \sum_{\beta=1}^{k_3} x_{k\alpha} \zeta_{\alpha\beta} y_{\alpha s\tau} \right| \le (r_3 + 1)\varepsilon.$$

Finally,

$$\begin{aligned}
\left| a_{ijk} - \sum_{l=1}^{r_1} \sum_{s=1}^{r_1} \sum_{t=1}^{r_2} \sum_{\tau=1}^{r_2} \sum_{\alpha=1}^{r_3} \sum_{\beta=1}^{r_3} (\phi_{ls} \psi_{t\tau} \zeta_{\alpha\beta} y_{\alpha t\tau}) u_{il} v_{jt} x_{k\alpha} \right| &\le \left| a_{ijk} - \sum_{s=1}^{r_1} \gamma_{is} z_{jks} \right| \\
+ \sum_{s=1}^{r_1} \left| z_{jks} - \sum_{t=1}^{r_2} \sum_{\tau=1}^{r_2} \psi_{t\tau} v_{jt} w_{ks\tau} \right| &+ \sum_{s=1}^{r_1} \sum_{\tau=1}^{r_2} \left| w_{ks\tau} - \sum_{\alpha=1}^{k_3} \sum_{\beta=1}^{k_3} x_{k\alpha} \zeta_{\alpha\beta} y_{\alpha s\tau} \right| \\
&\le (r_1 + 1)\varepsilon + r_1(r_2 + 1)\varepsilon + r_1 r_2 (r_3 + 1)\varepsilon,
\end{aligned}$$

which completes the proof. □

If $r_1 = r_2 = r_3 = r$, then the error bound becomes $(r+1)(r^2+r+1)\varepsilon \le (r+1)^3\varepsilon$. In the general case, we are not completely satisfied with the error bound of this theorem because it is not a symmetric function of r_1, r_2, r_3. Of course, the answer can be formally symmetrized, using different permutations of modes (for example, $n_3 \times n_2 \times n_1$) and taking the minimum of all these error bounds, but the obtained result seems to be rather artificial. So here we note that a "truly symmetric" version of this theorem is likely to need a different technique.

COROLLARY 3.2. [illegible]

$$||\mathcal{E}'||_F \le (r_1 r_2 r_3 + 2r_1 r_2 + 2r_1 + 1)\sqrt{n_1 n_2 n_3}\varepsilon.$$

4. The cross approximation method. For presentation purposes from now on we will assume that $n_1 = n_2 = n_3 = n$ and $r_1 = r_2 = r_3 = r$.

4.1. The two-dimensional-cross method. In the works [13, 14, 17] the problem of finding a rank-r approximation to a given matrix was connected with finding in matrix A a [illegible] (that is, determinant in modulus) among all $r \times r$ submatrices. The latter problem is hard to solve. However, we may be satisfied with a "sufficiently good" submatrix and some heuristic algorithms. Since these algorithms are to fetch a cross of some columns and rows, we call them [illegible]

[2]The matrix A is said to have ε-rank r if there exists a rank-r matrix B such that $\|A - B\| \le \varepsilon$.

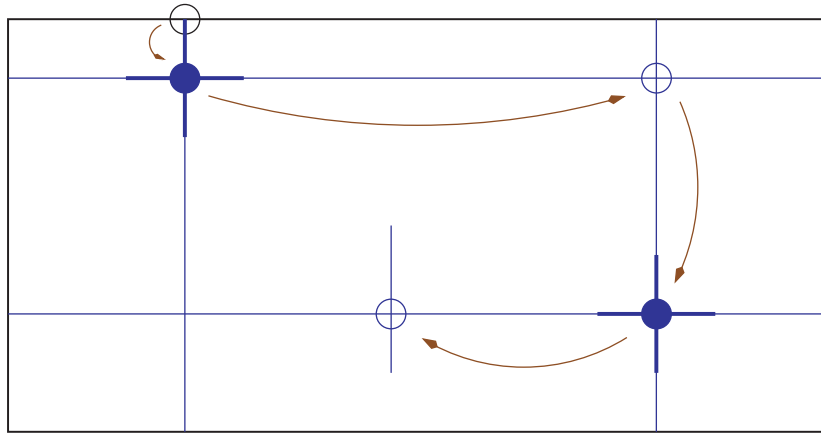


FIG. 1. *How a cross method works. Filled dots: elements used for the calculation of cross*(i_p, j_p). *Empty dots: row–pivot, step* (2).

. Probably the most simple and effective cross algorithm is the Gauss elimination method using some pivoting technique over dynamically selected sets of the entries of the "active matrix" (for a general description, see [9]). We will use here the column and row pivoting considered in [8]. This method is simple but may breakdown (quit when a good approximation is not obtained) if applied as it is. A cheap practical remedy proposed in [16] is a restarted version of this cross method. For the readers convenience, we give here a brief description of the algorithm.

ALGORITHM 1 (CROSS2D). Given a matrix A of approximate rank r, approximate it by a matrix $\tilde{A}_r$, which is a sum of r rank-1 matrices $u_p v_p^\top$ (so-called skeletons). The principle scheme is given in Figure 1.

(0) Numbering the steps by p, set $p = 1$. Choose some column in A, and assign its index to j_p.
(1) Calculate column j_p of the matrix A, and subtract from all elements the corresponding elements of already calculated skeletons. In the resulting vector find the largest magnitude element. Suppose it is located in the row i_p.
(2) Calculate the row i_p of the residue and the next pivot which is its largest magnitude element with a restriction that the element from the j_pth column cannot be chosen again (see Figure 1). Suppose this pivot is located in the j_{p+1}th column.
(3) Calculate the new cross centered at (i_p, j_p).
(4) If a stopping criterion is not satisfied, set $p := p + 1$, and go to step (1).

The approximation $\tilde{A}_p = \sum_{\alpha=1}^{p} u_\alpha v_\alpha^\top$ is considered good if

$$\|A - \tilde{A}_p\| \le \varepsilon \|A\|_F \approx \varepsilon \|\tilde{A}_p\|_F.$$

However, the exact computation of the error requires all matrix elements and n^2 operations, which is unacceptable. At the same time, the norm $\|\tilde{A}_p\|_F$ can be computed via the formula

$$\|\tilde{A}_p\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\sum_{\alpha=1}^{r} u_{i\alpha} v_{j\alpha}\right)^2 = \sum_{\alpha=1}^{r}\sum_{\alpha'=1}^{r}(u_\alpha, u_{\alpha'})(v_\alpha, v_{\alpha'})$$

using $\mathcal{O}(p^2 n)$ operations. And as a practical estimator of the error (stopping criterion),

we use the norm of a newly computed rank-1 correction. Specifically, we stop if

$$(n-p)\|u_p\|_2\|v_p\|_2 \le \varepsilon\|\tilde{A}_r\|_F.$$

The number $n-p$ is a heuristic constant. Note that after p steps of the cross algorithm exactly p rows and columns of the residue are zeroed, so if we assume that the error is "equally distributed" among the remaining $n-p$ rows, then we immediately arrive at the presented stopping criteria.

Such version of the cross method requires $2rn$ evaluations of matrix elements and $\mathcal{O}(r^2n)$ additional operations (the reason for counting the number of element computations is that the calculation of one element may be a very time-consuming operation). Even if the stopping criteria is satisfied, in some cases the obtained approximation is not good enough (but this does not happen very often). To make the method more robust, the [illegible] step is performed: we create a sample from the elements of the residue matrix $A - \tilde{A}_r$. If the error estimated from that sample is large, we proceed with step (3) using the largest magnitude element in the sample as a pivot.

4.2. Towards the three-dimensional-cross method. Consider the unfoldings of the array $\mathcal{A}$ (rectangular matrices of sizes $n \times n^2$ defined by (2.1)), and apply to them the cross approximation algorithm. If the array $\mathcal{A}$ possesses a good Tucker rank-(r, r, r) approximation, then there exist rank-r approximations for the unfoldings $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ which are also good:

$$\tilde{A}^{(1)}_r = U\Psi^\top, \qquad \tilde{A}^{(2)}_r = V\Phi^\top, \qquad \tilde{A}^{(3)}_r = W\Upsilon^\top,$$

where U, V, W are $n \times r$ matrices with [illegible] and matrices Ψ, Φ, Υ are $n^2 \times r$. The Tucker core is calculated by the convolution of the form (1.3) with a_{ijk} being replaced with their approximate values. For example, using the decomposition by the first direction, $\tilde{A}^{(1)}_r = U\Psi^\top$, we have

$$a_{ijk} \approx \tilde{a}_{ijk} = \sum_{\alpha=1}^{r} u_{i\alpha}\psi_{jk\alpha},$$

where the rows of the matrix $\Psi = [\psi_{(jk)\alpha}]$ are numbered by a pair of indices (jk). Substituting this into (1.3), we obtain

$$\begin{aligned} g_{i'j'k'} &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(\sum_{\alpha=1}^{r} u_{i\alpha}\psi_{jk\alpha}\right)u_{ii'}v_{jj'}w_{kk'} \\ &= \sum_{\alpha=1}^{r}(u_\alpha, u_{i'})\left(\sum_{j=1}^{n}\sum_{k=1}^{n} v_{jj'}w_{kk'}\psi_{jk\alpha}\right) \\ &= \sum_{j=1}^{n}\sum_{k=1}^{n} v_{jj'}w_{kk'}\psi_{jki'}. \end{aligned} \tag{4.1}$$

This computation needs $\mathcal{O}(n^2r)$ evaluations of the elements of $\mathcal{A}$ plus $\mathcal{O}(n^2r^2)$ operations.

Of course, $\mathcal{O}(n^2)$ is much smaller than the total number of elements in the array $\mathcal{A}$, but it is still too large when n is about 10^3.

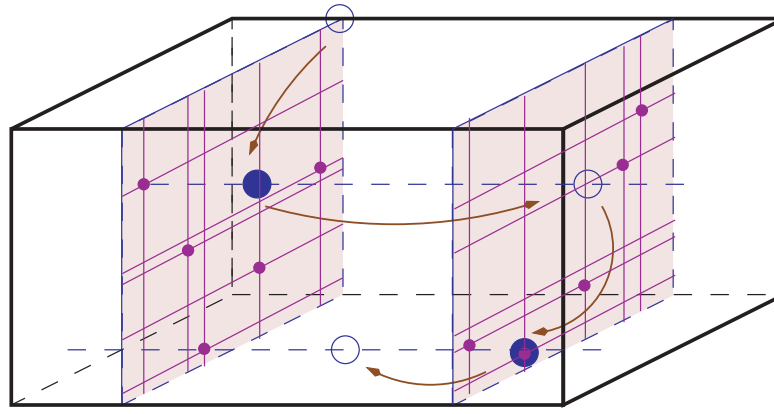


FIG. 2. *The work of the three-dimensional-cross method. The big filled and empty dots correspond to elements for the "outer" cross algorithm, and small dots show elements used for the "inner" cross algorithm, approximating a particular two-dimensional slice*

4.3. How to achieve linear complexity. We want to achieve linear complexity in n. To this end, we have to get rid of the computation of all elements in the slices of $\mathcal{A}$ used in the unfoldings (2.1) (then we avoid n^2-long vectors). We suggest to approximate the slices by the same cross algorithm developed for matrices. Since $\mathcal{A}$ has a good Tucker approximation with the accuracy ε, each slice $A_k = [(a_k)_{ij}]$ can be accurately approximated by a rank-r matrix. In what follows, we will never store a slice as a full $n \times n$ matrix and will never refer to all its elements. Instead, we deal only with some low-rank approximations for the slices.

ALGORITHM 2. Given an $n \times n \times n$ array $\mathcal{A}$, take one of the indices i, j, k as the "leading index," let it be k. Then consider the corresponding unfolding matrix of size $n \times n^2$, and approximate it applying the cross method. The columns of the unfolding matrix are calculated as usual, but each of the long rows is considered as a matrix of size $n \times n$ to be approximated by the same cross method. Therefore, at each step of the "outer" cross method we add one skeleton of form $A \otimes w$ with the long vector A represented by the sum of r skeletons in the "inner" cross method; therefore, we add a tensor of form $\sum_{q=1}^{r} u_q \otimes v_q \otimes w$. After r steps the approximant has the form

$$\tilde{A} = \sum_{p=1}^{r} \left(\sum_{q=1}^{r} u_q^{\{p\}} \otimes v_q^{\{p\}} \right) \otimes w^{\{p\}},$$

where $u_q^{\{p\}}$ and $v_q^{\{p\}}$, $q = 1, \ldots, r$, are vectors comprising the skeleton approximation to the long vector at the step with number p. Note that unlike vectors u and v, vectors w are numbered only by p, because k is the leading index. The principle scheme is given in Figure 2.

The expected number of arithmetic operations is [illegible] in sizes of the array: the complexity is $\mathcal{O}(nr^d)$ operations for some small $d > 0$.

(0) Numbering the steps by p, set p to 1. Choose a slice $A_k = [(a_k)_{ij}]$ in $\mathcal{A}$, for example, and assume its index to k_1. Set $\tilde{\mathcal{A}} = 0$.

(1a) Find an approximation A_{k_p} to the k_pth slice of the residue $\mathcal{R} = \mathcal{A} - \tilde{\mathcal{A}}_p$ by the cross method:

$$A_{k_p} = \sum_{q=1}^{r} u_q v_q^{\top}.$$

(1b) Find the largest magnitude element in the matrix A_{k_p}; let it be located at (i_p, j_p).

(2) Compute the vector w corresponding to the fiber of $\mathcal{R}$ with index (i_p, j_p),

$$w_k = \mathcal{R}_{i_p, j_p, k},$$

perform the scaling

$$w := w / w_{k_p},$$

and find in w the largest magnitude element from those whose index is not equal to k_p.[3] Suppose it is located at the k_{p+1}th position of w.

(3) Compute a new approximation:

$$\tilde{\mathcal{A}} := \tilde{\mathcal{A}} + A_{k_p} \otimes w = \tilde{\mathcal{A}} + \left(\sum_{q=1}^{r} u_q v_q^\top \right) \otimes w = \tilde{\mathcal{A}} + \sum_{q=1}^{r} u_q \otimes v_q \otimes w.$$

(4) If the stopping criterion is not satisfied, set $p := p + 1$, and go to step (1).

In the end, the array $\mathcal{A}$ is approximated by $\tilde{\mathcal{A}} = [\tilde{a}_{ijk}]$ having a Tucker-like decomposition (also viewed as a trilinear, PARAFAC, or CANDECOMP decomposition [1, 5]) of the form (in elementwise notation)

$$\tilde{a}_{ijk} = \sum_{p=1}^{r} \left(\sum_{q=1}^{r} u_{iq}^{\{p\}} v_{jq}^{\{p\}} \right) w_k^{\{p\}} = \sum_{\alpha=1}^{r^2} u_{i\alpha} v_{j\alpha} w_{k\alpha}, \tag{4.2}$$

where in the second sum for simplicity all terms are numbered by a single index α instead of a complicated sum on the left.

During the implementation of this method, we encounter several problems that should be solved with a linear complexity in n:

- determine the largest magnitude element in a low-rank matrix, step (1)b;
- estimate the quantities in the relationships

$$\|\mathcal{A} - \tilde{\mathcal{A}}\|_F \le \varepsilon \|\mathcal{A}\|_F \approx \varepsilon \|\tilde{\mathcal{A}}\|_F$$

so as to have a sound stopping criterion, step (4).

The first problem is not trivial, and we do not know if there is an exact and fast way to find a maximal element in a low-rank matrix. However, we are able to design a heuristic algorithm, based on the submatrix of maximal volume. It manifests a very good practical performance (see Appendix A).

The stopping criterion in the Cross3D method is identical to the two-dimensional case, by the comparison of the approximant norm and the norm of a newly computed cross-correction. The norm $\|\tilde{\mathcal{A}}\|_F$ is computed by the formulas

$$\begin{aligned} \|\tilde{\mathcal{A}}\|_F^2 &= \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \left(\sum_{\alpha=1}^{r^2} u_{i\alpha} v_{j\alpha} w_{k\alpha} \right)^2 \\ &= \sum_{\alpha=1}^{r^2} \sum_{\alpha'=1}^{r^2} (u_\alpha, u_{\alpha'})(v_\alpha, v_{\alpha'})(w_\alpha, w_{\alpha'}). \end{aligned}$$

[3] It is worthy to note that we cannot also use elements with indices $k_1, \ldots, k_{p-1}$, but it can be verified that they are all zeroes, so they cannot have maximal modulus.

The cost of Algorithm 2 is $\mathcal{O}(nr^2)$ evaluations of the elements of $\mathcal{A}$ plus $\mathcal{O}(nr^4)$ arithmetic operations (at each outer step of the method we compute a new slice from which we should subtract the elements of the previously computed approximation, and that results in a relatively big constant r^4 at the size n). This is already a linear complexity. However, we are going to present a "clever" implementation with a significantly better performance.

4.4. The three-dimensional-cross algorithm. We can improve the efficiency of Algorithm 2 by using a more compact way to store and handle the slices A_{k_p} so that the required number of vectors to represent them is reduced from $\mathcal{O}(r^2)$ to $\mathcal{O}(r)$.

At each step we approximate the computed slices A_{k_p} in the format

$$A_{k_p} = UB_pV^\top, \tag{4.3}$$

where $n \times r$ matrices U, V are orthogonal and the core matrices B_p are $r \times r$. It is worth noting that (4.3) is also known as a Tucker2 decomposition, where only 2 of 3 modes are compressed. The storage for p slices is now $2nr + pr^2$ which is asymptotically equal to $\mathcal{O}(nr)$. The existence of matrices U, V, follows from the existence of a "good" Tucker approximation. In fact, we can try U and V as the Tucker factors. The computation of this simultaneous matrix decomposition is equivalent to the computation of the Tucker decomposition of a $n \times n \times p$ array:

$$\mathcal{A}' = [A_{k_1} \dots A_{k_p}].$$

Indeed, if

$$a_{ijk_p} = \sum_{i'=1}^{r} \sum_{j'=1}^{r} \sum_{p'=1}^{r} g_{i'j'p'} u_{ii'} v_{jj'} w_{pp'},$$

then, setting

$$\sum_{p'=1}^{r} g_{i'j'p'} w_{pp'} = b_{i'j'p} = (b_p)_{i'j'},$$

we immediately arrive at (4.3).

Another important modification concerns the computation of the slices. Suppose the p steps are done and we are going to compute the $p+1$th slice $A_{k_{p+1}}$. Instead of using the "full" cross method for this slice, we first find an approximation of the form

$$A_{k_{p+1}} \approx U\Phi V^\top,$$

where U, V come from (4.3) and a matrix Φ is $r \times r$. Such an approximation can be obtained quite cheaply by the following scheme:

- Find $r \times r$ [illegible] in U and V. Suppose they have indices $i_1, \dots, i_r$ and $j_1, \dots, j_r$. Denote these submatrices by $\hat{U}$ and $\hat{V}$.
- Compute the $r \times r$ submatrix S in $A_{k_{p+1}}$ lying on the intersection of rows with indices $i_1, \dots, i_r$ and columns with indices $j_1, \dots j_r$.
- Compute

$$\Phi = \hat{U}^{-1} S \hat{V}^{-1}. \tag{4.4}$$

We can prove that this approximation approach is robust (see Appendix B). After Φ is computed, we check the approximation error by taking some random samples of a true matrix A_{k_p}. If the approximation is not good enough, then we perform some steps of the cross approximation algorithm, starting from a good approximation. However, as a rule, only a few steps (or even none) of the cross algorithm are required.

ALGORITHM 3 (CROSS3D). Suppose an $n \times n \times n$ three-way array $\mathcal{A}$ is given.

(0) Perform one step of Algorithm 2 (with $p = 1$). Upon completion, $p = 2$, and $\mathcal{A}$ is represented as

$$\tilde{A} = \left(\sum_{q=1}^{r} u_q^{\{1\}} (v_q^{\{1\}})^\top \right) \otimes w^{\{1\}} = \sum_{q=1}^{r} u_q^{\{1\}} \otimes v_q^{\{1\}} \otimes w^{\{1\}}.$$

Form matrices $U_1 = [u_q^{\{1\}}]$, $V_1 = [v_q^{\{1\}}]$, and compute orthonormal bases U, V of these subspaces using two QR-decompositions:

$$U_1 = UR_u, \qquad V_1 = VR_v.$$

Then $\mathcal{A}$ is represented as

$$\tilde{A} = (UR_uR_v^\top V^\top) \otimes w^{\{1\}} = (UB_1V^\top) \otimes (w^{\{1\}}/\|w^{\{1\}}\|_2),$$

$$B_1 = R_uR_v^\top \|w^{\{1\}}\|_2.$$

Set $w^{\{1\}} := w^{\{1\}}/\|w^{\{1\}}\|$. (Note that in this algorithm we will keep bases U, V, and W orthonormal.) In the vector $w^{\{1\}}$ compute the largest magnitude element; suppose it has index k_2.

(1.1) Compute Φ from (4.4). If necessary, perform some additional steps of the cross method to obtain an approximation to the slice A_{k_p}:

$$A_{k_p} \approx \tilde{A}_{k_p} = U\Phi V^\top + \sum_{q=1}^{r_p} u_q^{\{p\}} (v_q^{\{p\}})^\top.$$

(Note that r_p is supposed to be small even compared to r.)

(1.2) Add new vectors $U_p = [u_q^{\{p\}}]$, $V_p = [v_q^{\{p\}}]$, $q = 1, \ldots, r$, to bases U, V, and orthogonalize the extended matrices $[UU_p]$, $[VV_p]$:

$$[UU_p] = [U\hat{U}_p] \begin{bmatrix} I & M_u \\ 0 & R_u \end{bmatrix}, \quad [VV_p] = [V\hat{V}_p] \begin{bmatrix} I & M_v \\ 0 & R_v \end{bmatrix},$$

$$U^\top \hat{U}_p = 0, \qquad V^\top \hat{V}_p = 0, \qquad \hat{U}_p^\top \hat{U}_p = I, \qquad \hat{V}_p^\top \hat{V}_p = I.$$

(1.3) Compute new $(r + r_p) \times (r + r_p)$ core B_p:

$$B_p = \begin{bmatrix} M_u \\ R_u \end{bmatrix} \begin{bmatrix} M_v^\top R_v^\top \end{bmatrix}.$$

Other slices in a new basis have the form

$$A_{k_q} = [U\hat{U}_p] \begin{bmatrix} B_q & 0 \\ 0 & 0 \end{bmatrix} [V\hat{V}_p]^\top, \qquad q = 1, \ldots, p-1.$$

Therefore, approximation $\tilde{\mathcal{A}}_{p-1}$ is represented as

$$\tilde{\mathcal{A}}_{p-1} = \sum_{q=1}^{p-1} \left(U' B'_q V'^{\top}\right) \otimes w_q,$$

$$U' = [U\hat{U}_p], \qquad V' = [V\hat{V}_p], \qquad B'_q = \begin{bmatrix} B_q & 0 \\ 0 & 0 \end{bmatrix}, \qquad q = 1, \ldots, p-1.$$

Set also $B'_p = B_p$.

(1.4) In the new slice A_{k_p} the largest magnitude element is found. Suppose it is located in (i_p, j_p).

(2.1) The fiber $w^{\{p\}}$ of the residue $\mathcal{A} - \tilde{\mathcal{A}}_{p-1}$ corresponding to (i_p, j_p) is computed.

(2.2) Vector $w^{\{p\}}$ is orthogonalized to vectors $W = [w^{\{1\}}, \ldots, w^{\{p-1\}}]$:

$$w^{\{p\}} = \sum_{q=1}^{p-1} c_q w^{\{q\}} + \hat{w}^{\{p\}}, \qquad (\omega^{\{q\}})^{\top} \hat{w}^{\{p\}} = 0, \qquad q = 1, \ldots, p-1.$$

Cores of old slices B'_q, $q = 1, \ldots, p-1$, are modified:

$$B''_q = B'_q + c_q B'_p, \qquad q = 1, \ldots, p-1.$$

Vector $\hat{w}^{\{p\}}$ is normalized:

$$w^{\{p\}} := \hat{w}^{\{p\}} / \|\hat{w}^{\{p\}}\|_2, \qquad B''_p = \|\hat{w}^{\{p\}}\|_2 B'_p.$$

(3) The approximation $\tilde{\mathcal{A}}_p$ is represented as

$$\tilde{\mathcal{A}}_p = \sum_{q=1}^{p} \left(U' B''_q V'^{\top}\right) \otimes w^{\{q\}}. \tag{4.5}$$

To reduce the sizes of the $(r + r_p) \times (r + r_p)$ matrices B''_q, we apply the Tucker reduction method.

(3.1) Create a three-way array $\mathcal{B}'' = [B''_1 \ldots B''_p]$ of size $(r + r_p) \times (r + r_p) \times p$, and compute its Tucker decomposition.

$$b''_{ijk} = \sum_{i'=1}^{r} \sum_{j'=1}^{r} \sum_{k'=1}^{r} g^{\clubsuit}_{i'j'k'} u^{\clubsuit}_{ii'} v^{\clubsuit}_{jj'} w^{\clubsuit}_{kk'}.$$

If we introduce matrices $B_{\clubsuit k}$ with elements

$$(B_{\clubsuit k})_{i'j'} = \sum_{k'=1}^{r} g^{\clubsuit}_{i'j'k'} w^{\clubsuit}_{kk'},$$

then we have

$$B''_k = U_{\clubsuit} B_{\clubsuit k} V_{\clubsuit}^{\top}, \qquad k = 1, \ldots, p, \tag{4.6}$$

where matrices $U_{\clubsuit}$ and $V_{\clubsuit}$ are $(r + r_1) \times r$ and cores $B_{\clubsuit k}$ are $r \times r$.

(3.2) Substituting (4.6) into (4.5), we obtain that

$$\tilde{\mathcal{A}}_p = \sum_{k=1}^{p} \Big((U' U_{\clubsuit}) B_{\clubsuit_k} (V' V_{\clubsuit})^\top \Big) \otimes w^{\{k\}} = \sum_{k=1}^{p} \big(U B_k V^\top \big) \otimes w^{\{k\}}, \tag{4.7}$$

where $U = U' U_{\clubsuit}$, $V = V' V_{\clubsuit}$ are orthogonal and cores $B_k = B_{\clubsuit_k}$ are $r \times r$. The format (4.3) is restored.

(4) Check stopping criteria; if it is not satisfied, go to (1.1).

At each step of the outer cross method we obtain a new Tucker approximant with the core of the dimension $(r + r_p) \times (r + r_p) \times r$; the purpose of steps (3.1)–(3.2) is to compress this tensor back.

The Algorithm 3 is the final version of Cross3D. The numerical complexity of the method is $\mathcal{O}(nr)$ evaluations of the array elements and $\mathcal{O}(nr^3)$ additional operations.

5. Numerical experiments. We illustrate the performance of our algorithm on some model tensors which allow good low-rank approximation.

Specifically, we consider the following two types of arrays:

$$\mathcal{A} = [a_{ijk}], \qquad a_{ijk} = \frac{1}{i + j + k}, \qquad 1 \le i, j, k \le n,$$

$$\mathcal{B} = [b_{ijk}], \qquad b_{ijk} = \frac{1}{\sqrt{i^2 + j^2 + k^2}}, \qquad 1 \le i, j, k \le n.$$

The rank estimates obtained in [15, 18, 19] have the form

$$r \le C(\log n \log^2 \varepsilon),$$

where ε is an error of the approximation, so the rank grows only logarithmically with n and ε.

These two examples arise from the numerical solution of integral equations. For example, the array $\mathcal{B}$ is obtained from the integral equation with kernel $\frac{1}{||x-y||}$ acting on a unit cube and being disretized by the Nyström method on a uniform grid.

Table 1 shows the ranks, accuracies, and size of the computed Tucker approximation for the array $\mathcal{A}$; Table 2 shows the same for $\mathcal{B}$. The accuracy of the approximation was computed by sampling the elements of the array, since it is not possible to check all the elements for large n. The size of the sample was determined by the following rule: if the sample size was doubled, the estimated error should change by no more than 10%. As it can be seen, the approximation method is robust and leads to astonishing memory savings: the arrays that would need in the full format an enormous storage of 2 petabytes ($2 \cdot 2^{50}$ PB) are compressed to the sizes of 100 MB.

Moreover, our algorithm works with arrays on this huge scale on a personal workstation. The timings made on a personal computer (`Pentium-4` with 3.4 Ghz clock) are shown in Figure 3. This figure confirms that the approximation time is almost linear in n. More precisely, we demonstrate the $\mathcal{O}(nr^3)$ complexity of the Cross3D algorithm with rank estimated as $r \sim \log n$ for fixed ε. Therefore, the complexity of the algorithm is estimated as

$$t \le cn \log^3 n.$$

In Figure 3 real timings, measured for different ε, are shown together with "theoretical" bounds $cn \log^3 n$, plotted for two different values of c. The somewhat irregular

TABLE 1
Numerical results

$$\mathcal{A} = [a_{ijk}], \qquad a_{ijk} = \frac{1}{i+j+k}, \qquad 1 \le i,j,k \le n.$$

Rank and accuracy of the decomposition.

ε	1.10−3		1.10−5		1.10−7		1.10−9	
n	r	err	r	err	r	err	r	err
64	5	$2.57_{10}-4$	8	$2.33_{10}-6$	10	$1.46_{10}-8$	12	$3.09_{10}-10$
128	6	$6.86_{10}-4$	8	$4.47_{10}-6$	11	$3.19_{10}-8$	13	$6.4_{10}-10$
256	6	$8.84_{10}-4$	9	$3.97_{10}-6$	12	$5.49_{10}-8$	15	$3.03_{10}-10$
512	7	$7.49_{10}-4$	10	$1.41_{10}-6$	13	$6.84_{10}-8$	16	$5.2_{10}-10$
1024	7	$5.71_{10}-4$	11	$4.83_{10}-6$	14	$4.09_{10}-8$	18	$2.74_{10}-10$
2048	7	$6.63_{10}-4$	12	$2.08_{10}-6$	16	$4.01_{10}-8$	19	$3.97_{10}-10$
4096	8	$3.23_{10}-4$	12	$6.32_{10}-6$	17	$3.44_{10}-8$	21	$3.47_{10}-10$
8192	8	$6.36_{10}-4$	13	$3.36_{10}-6$	18	$1.93_{10}-8$	22	$4.56_{10}-10$
16384	9	$7.95_{10}-4$	14	$3.52_{10}-6$	19	$7.21_{10}-8$	24	$5.64_{10}-10$
32768	9	$6.4_{10}-4$	14	$8.86_{10}-6$	20	$5.27_{10}-8$	25	$3.79_{10}-10$
65536	9	$4.07_{10}-4$	15	$6.31_{10}-6$	21	$2.51_{10}-8$	26	$5.25_{10}-10$

Rank and size (MB) of the Tucker decomposition.
The sizes smaller than 1MB are not shown.

ε		1.10−3		1.10−5		1.10−7		1.10−9	
n	full	r	mem	r	mem	r	mem	r	mem
64	2MB	5		8		10		12	
128	16MB	6		8		11		13	
256	128MB	6		9		12		15	
512	1GB	7		10		13		16	
1024	8GB	7		11		14		18	
2048	64GB	7		12		16		19	
4096	512GB	8	< 1	12	1.1	17	1.6	21	2
8192	4TB	8	1.5	13	2.5	18	3.5	22	4.2
16384	32TB	9	3.4	14	5.25	19	7.2	24	9
32768	256TB	9	6.75	14	10.5	20	15	25	19
65536	2PB	9	13.5	15	22	21	31	26	39

behavior on the timing plots is caused by the effects of caching (for small n) and by some rank overestimation by the stopping criteria for large n.

Two dense tensors considered come from a simple disretization of integral equations. Despite their "regularity" they are quite representative: in more complex cases our method behaves similarly. In other areas where tensor decomposition is used, the researchers often obtain more irregular and possibly sparse tensors. We want to note that sparseness is [illegible] for Cross3D because in that case the residue can be measured [illegible] and the pivots during the cross approximation stage can be also found exactly, leading to a [illegible] method. The applications of the three-dimensional-cross approach to more complex tensors will be reported elsewhere.

Appendix A. How to find the maximal element in a slice. One of the important ingredients of the three-dimensional-cross method is the determination of the maximal element in a given low-rank matrix in linear time.

Suppose we have computed a skeleton approximation to a low-rank matrix

$$A = UV^{\top},$$

where U, V are $n \times r$, and we want to find the largest magnitude element in it. This

TABLE 2
Numerical results

$$\mathcal{B} = [b_{ijk}], \qquad b_{ijk} = \frac{1}{\sqrt{i^2 + j^2 + k^2}}, \qquad 1 \le i, j, k \le n.$$

Rank and accuracy of the Tucker decomposition.

ε	1.10−3		1.10−5		1.10−7		1.10−9	
n	r	err	r	err	r	err	r	err
64	7	$3.77_{10}-4$	11	$3.91_{10}-6$	14	$5.7_{10}-8$	18	$2.21_{10}-10$
128	8	$5.19_{10}-4$	12	$5.92_{10}-6$	17	$2._{10}-8$	20	$5.63_{10}-10$
256	9	$4.11_{10}-4$	14	$6.4_{10}-6$	19	$3.46_{10}-8$	23	$4.5_{10}-10$
512	10	$4.93_{10}-4$	15	$6.67_{10}-6$	21	$2.92_{10}-8$	26	$3.27_{10}-10$
1024	10	$5.47_{10}-4$	17	$3.21_{10}-6$	23	$3.95_{10}-8$	29	$4.73_{10}-10$
2048	11	$4.98_{10}-4$	18	$5.26_{10}-6$	25	$6.83_{10}-8$	31	$5.94_{10}-10$
4096	12	$8.4_{10}-4$	19	$4.25_{10}-6$	27	$3.56_{10}-8$	34	$3.38_{10}-10$
8192	12	$6.8_{10}-4$	20	$6._{10}-6$	28	$5.8_{10}-8$	36	$3.66_{10}-10$
16384	13	$2.69_{10}-4$	22	$4.78_{10}-6$	31	$5.65_{10}-8$	39	$2.67_{10}-10$
32768	13	$8.52_{10}-4$	23	$6.09_{10}-6$	32	$7.16_{10}-8$	41	$5.51_{10}-10$
65536	14	$6.27_{10}-4$	24	$6.52_{10}-6$	34	$7.89_{10}-8$	44	$1.41_{10}-9$

Rank and size (MB) of the Tucker decomposition.
Values less than 1MB are not shown

ε		1.10−3		1.10−5		1.10−7		1.10−9	
n	full	r	mem	r	mem	r	mem	r	mem
64	2MB	7		11		14		18	
128	16MB	8		12		17		20	
256	128MB	9		14		19		23	
512	1GB	10		15		21		26	
1024	8GB	10		17		23		29	
2048	64GB	11	< 1	18	< 1	25	1.18	31	1.46
4096	512GB	12	1.15	19	1.78	27	2.54	34	3.2
8192	4TB	12	2.25	20	3.75	28	5.3	36	6.8
16384	32TB	13	4.9	22	8.25	31	11.7	39	14.7
32768	256TB	13	9.75	23	17.25	32	24	41	31
65536	2PB	14	21	24	36	34	51	44	66

problem can be solved by comparing all the elements of the matrix, but it costs $\mathcal{O}(n^2)$ operations. The proposed algorithm is based on the following hypothesis.

Consider $r \times r$ submatrices in a rank-r matrix A. Let B be a submatrix of maximal volume among all such submatrices. Then

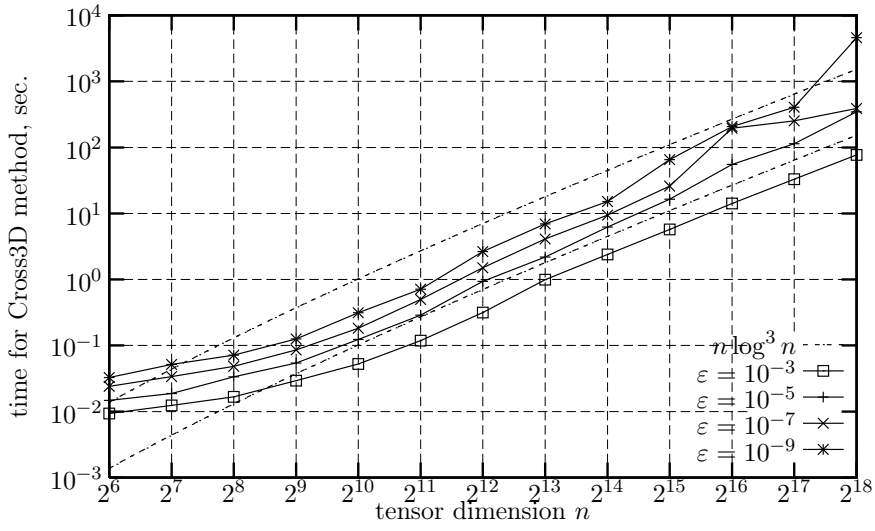
$$||B||_C \ge \frac{||A||_C}{r}.$$

So the maximal element in the submatrix of maximal volume cannot be very much different from the maximal element in the whole matrix A.

How does one determine a submatrix of maximal volume? This submatrix of A lies on the intersection of rows $i_1, \ldots, i_r$, coinciding with rows which contain the submatrix of maximal volume in U, and columns $j_1, \ldots, j_r$, which contain the submatrix of maximal volume in $V^\top$. To find the submatrix of maximal volume in a $n \times r$ matrix, we will use the algorithm, proposed in [12]. For the readers convenience we describe it below.

ALGORITHM 4. Suppose U is a $n \times r$ matrix and its $r \times r$ submatrix with maximal volume is needed.

$$\mathcal{A} = [a_{ijk}], \qquad a_{ijk} = \frac{1}{i+j+k}, \qquad 1 \le i,j,k \le n$$



$$\mathcal{B} = [b_{ijk}], \qquad b_{ijk} = \frac{1}{\sqrt{i^2+j^2+k^2}}, \qquad 1 \le i,j,k \le n$$

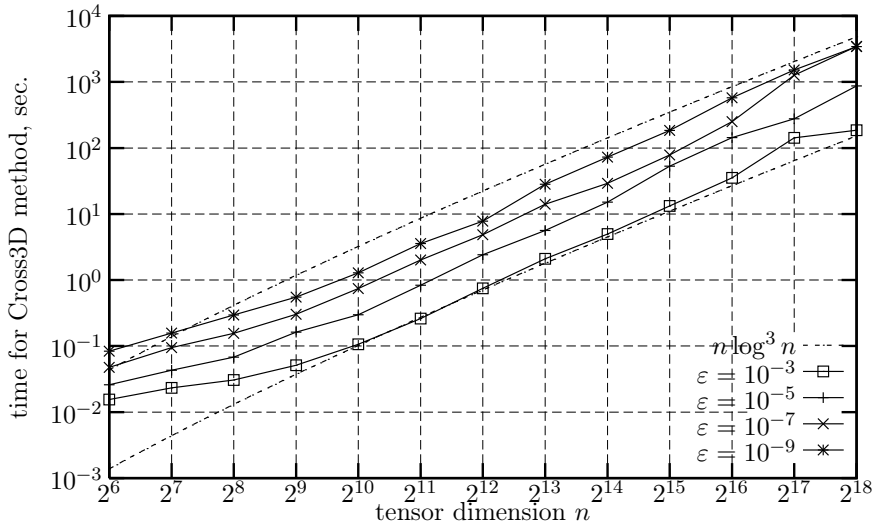


FIG. 3. *Approximation time, sec.*

(0) Let A_γ be a leading submatrix. In the beginning set A_γ to any nonsingular submatrix of A, and permute the rows so that A_γ is located in the first r rows.

(1) Compute

$$AA_\gamma^{-1} = \begin{bmatrix} I_{r\times r} \\ Z \end{bmatrix} = B.$$

(2) Find the largest magnitude element $|z_{ij}|$ in Z.
(3) **If** $\gamma = |z_{ij}| > 1$, **then**
Permute in B rows $r+i$ and j. The upper submatrix in B after the permutation has the form

$$\begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ * & * & \gamma & * & * & \\ & & & \ddots & & \\ & & & & 1 & \end{pmatrix},$$

and its determinant is equal to $\gamma \geq 1+\varepsilon$, that is, it increased. Denote by A_γ the new submatrix in the first r rows of A, and return to step (1).
Otherwise terminate the algorithm.

In practice, to avoid a huge number of transpositions a more "soft" stopping criteria is used in step (3). The algorithm stops if $|z_{ij}| \leq 1+\nu$, where ν is a some small parameter.

Appendix B. The $U\Phi V^\top$ decomposition. In this appendix we will prove that the usage of (4.4) for the construction of the low-rank approximation to a slice is "legal."

THEOREM B.1. [illegible] A [illegible] $n_1 \times n_2$ [illegible] U, V [illegible] $n_1 \times r_1$ [illegible] $n_2 \times r_2$ [illegible] Φ [illegible]

$$A = U\Phi V^\top + E, \quad ||E||_F \leq \varepsilon.$$

[illegible] Φ' [illegible] (4.4) [illegible]

$$||A - U\Phi' V^\top||_F \leq (\sqrt{n_1 r_1 n_2 r_2} + 1)\varepsilon.$$

[illegible]. If $\hat{U}$ and $\hat{V}$ are submatrices of maximal volume in U and V, respectively, and $\hat{A}$ is a submatrix in A lying on the intersection of the selected rows from U and columns from $V^\top$, then

$$\hat{A} = \hat{U}\Phi\hat{V}^\top + \hat{E},$$

where $\hat{E}$ is a submatrix of E occupying the same positions in E as $\hat{A}$ in A.

$$||\Phi - \Phi'|| \leq ||\hat{U}^{-1}||\ ||\hat{E}||\ ||\hat{V}^{-1}||.$$

The norms $\hat{U}^{-1}, \hat{V}^{-1}$ can be estimated as follows. We know that the elements of

$$U\hat{U}^{-1}$$

are not greater than 1 in modulus (because $\hat{U}$ is a submatrix of maximal volume). Therefore,

$$||\hat{U}^{-1}||_F \leq \sqrt{n_1 r_1}.$$

Using this estimate we immediately complete the proof as follows:

$$\|A - U\Phi' V^\top\|_F \leq \|U\Phi V^\top - U\Phi' V^\top\|_F + \|E\|_F = \|\Phi - \Phi'\| + \varepsilon \leq \sqrt{n_1 r_1}\sqrt{n_2 r_2}\varepsilon + \varepsilon. \qquad \square$$

Acknowledgment. We are very grateful to both of the referees of our paper. The remark of one of the referees helped us to discover a nasty bug in the program code.

REFERENCES

[1] R. A. Harshman, *Foundations of the Parafac procedure: Models and conditions for an explanatory multimodal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.
[2] Three-Mode Company, http://three-mode.leidenuniv.nl.
[3] P. Comon, *Tensor decomposition: State of the art and applications*, in IMA Conference on Mathematics in Signal Processing, Warwick, UK, http://www.i3s.unice.fr/~comon/FichiersPs/ima2000.ps
[4] J. D. Caroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart–Young decomposition*, Psychometrika, 35 (1970), pp. 283–319.
[5] R. Bro, *PARAFAC: Tutorial and applications*, Chemom. Intelligent Lab. Systems, 38 (1997), pp. 149–171.
[6] G. Beylkin and M. M. Mohlenkamp, *Numerical operator calculus in higher dimensions*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 10246–10251.
[7] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear image analysis for facial recognition*, Proceedings of the International Conference on Pattern Recognition, Quebec City, Canada, 2000, pp. 511–514.
[8] M. Bebendorf, *Approximation of boundary element matrices*, Numer. Math., 86 (2000), pp. 565–589.
[9] J. M. Ford and E. E. Tyrtyshnikov, *Combining Kronecker product approximation with discrete wavelet transforms to solve dense, function-related systems*, SIAM J. Sci. Comput., 25 (2003), pp. 961–981.
[10] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.
[11] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On best rank-*1 *and rank-*$(R_1, R_2, \ldots, R_N)$ *approximation of high-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.
[12] S. A. Goreinov, *Pseudoskeleton approximations of block matrices generated by asymptotically smooth kernels*, Ph.D. Thesis, Institute on Numerical Mathematics, Russian Academy of Sciences, Moscow, Russia, 2001 (in Russian).
[13] S. A. Goreinov and E. E. Tyrtyshnikov, *The maximal-volume concept in approximation by low-rank matrices*, Contemp. Math., 208 (2001). pp. 47–51.
[14] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *A theory of pseudo-skeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21.
[15] W. Hackbusch, B. N. Khoromskij, and E. E. Tyrtyshnikov, *Hierarchical Kronecker tensor-product approximations*, J. Numer. Math., 13 (2005), pp. 119–156.
[16] D. V. Savostianov, *Mosaic-skeleton approximations*, Master Thesis, Institute on Numerical Mathematics, Russian Academy of Sciences, Moscow, Russia, 2001 (in Russian).
[17] E. E. Tyrtyshnikov, *Incomplete cross approximation in the mosaic-skeleton method*, Computing, 4 (2000), pp. 367–380.
[18] E. E. Tyrtyshnikov, *Kronecker-product approximations for some function-related matrices*, Linear Algebra Appl., 379 (2004), pp. 423–437.
[19] E. E. Tyrtyshnikov, *Tensor approximations of matrices generated by asymptotically smooth functions*, Sb. Math., 194 (2003), pp. 941–954.
[20] I. Ibraghimov, *Application of the three-way decomposition for matrix compression*, Numer. Linear Algebra Appl., 9 (2002), pp. 551–565.
[21] V. Olshevsky, I. V. Oseledets, and E. E. Tyrtyshnikov, *Tensor properties of multilevel Toeplitz and related matrices*, Linear Algebra Appl., 412 (2006), pp. 1–21.
[22] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
[23] P. Drineas, R. Kannan, and M. W. Mahoney, SIAM J. Comput., 36 (2006), pp. 158-183.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 957–987

TENSOR-CUR DECOMPOSITIONS FOR TENSOR-BASED DATA*

MICHAEL W. MAHONEY†, MAURO MAGGIONI‡, AND PETROS DRINEAS§

Abstract. Motivated by numerous applications in which the data may be modeled by a variable subscripted by three or more indices, we develop a tensor-based extension of the matrix CUR decomposition. The tensor-CUR decomposition is most relevant as a data analysis tool when the data consist of one mode that is qualitatively different from the others. In this case, the tensor-CUR decomposition approximately expresses the original data tensor in terms of a basis consisting of underlying subtensors that are actual data elements and thus that have a natural interpretation in terms of the processes generating the data. Assume the data may be modeled as a $(2+1)$-tensor, i.e., an $m \times n \times p$ tensor $\mathcal{A}$ in which the first two modes are similar and the third is qualitatively different. We refer to each of the p different $m \times n$ matrices as "slabs" and each of the mn different p-vectors as "fibers." In this case, the tensor-CUR algorithm computes an approximation to the data tensor $\mathcal{A}$ that is of the form $\mathcal{CUR}$, where $\mathcal{C}$ is an $m \times n \times c$ tensor consisting of a small number c of the slabs, $\mathcal{R}$ is an $r \times p$ matrix consisting of a small number r of the fibers, and $\mathcal{U}$ is an appropriately defined and easily computed $c \times r$ encoding matrix. Both $\mathcal{C}$ and $\mathcal{R}$ may be chosen by randomly sampling either slabs or fibers according to a judiciously chosen and data-dependent probability distribution, and both c and r depend on a rank parameter k, an error parameter ϵ, and a failure probability δ. Under appropriate assumptions, provable bounds on the Frobenius norm of the error tensor $\mathcal{A} - \mathcal{CUR}$ are obtained. In order to demonstrate the general applicability of this tensor decomposition, we apply it to problems in two diverse domains of data analysis: hyperspectral medical image analysis and consumer recommendation system analysis. In the hyperspectral data application, the tensor-CUR decomposition is used to *compress* the data, and we show that classification quality is not substantially reduced even after substantial data compression. In the recommendation system application, the tensor-CUR decomposition is used to *reconstruct* missing entries in a user-product-product preference tensor, and we show that high quality recommendations can be made on the basis of a small number of basis users and a small number of product-product comparisons from a new user.

Key words. CUR decomposition, tensor decomposition, hyperspectral imagery, recommendation system

AMS subject classifications. 15A23

DOI. 10.1137/060665336

1. Introduction.

Novel algorithmic methods to structure large data sets are of continuing interest. A particular challenge is presented by tensor-based data, i.e., data which are modeled by a variable subscripted by three or more indices [44, 31, 46, 61, 11]. Numerous examples suggest themselves, but to guide the discussion consider the following three. First, in internet data applications, if one is studying the properties of a large time-evolving graph, the data may consist of a graph or its adjacency matrix sampled at a large number of sequential time steps, in which case $\mathcal{A}_{ijk}$ may represent the weight of the edge between nodes i and j at time step k. Second, in biomedical

*Received by the editors July 17, 2006; accepted for publication (in revised form) by L. De Lathauwer January 8, 2007; published electronically September 25, 2008. A preliminary version of this paper appeared in *Proceedings of the 12th Annual ACM SIGKDD Conference*, 2006, pp. 327–336.
http://www.siam.org/journals/simax/30-3/66533.html

†Yahoo Research, Sunnyvale, CA 94089 (mahoney@yahoo-inc.com). Part of this work was performed while this author was at the Department of Mathematics, Yale University, New Haven, CT 06520.

‡Department of Mathematics, Duke University, Durham, NC 27708 (mauro.maggioni@duke.edu). Part of this work was performed while this author was at the Department of Mathematics, Yale University, New Haven, CT 06520. This author's research was partially supported by NSF-DMS grant 0512050.

§Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180 (drinep@cs.rpi.edu).

data applications, if one is studying cancer diagnosis, the data may consist of a large number of hyperspectrally resolved biopsy images, in which case $\mathcal{A}_{ijk}$ may represent the absorbed or transmitted light intensity of a biopsy sample at pixel ij at frequency k. Third, in consumer data applications, if one is studying recommendation systems, the data may consist of product-product preference data for a large number of users, in which case $\mathcal{A}_{ijk}$ may be ± 1, depending on whether product i or j is preferred by user k. Tensor-based data are particularly challenging due to their size and since many data analysis tools based on graph theory and linear algebra do not easily generalize.

When compared with algorithmic results for data modeled by either matrices or graphs, algorithmic results for data modeled by multimode tensors are modest. For example, even computing the rank of a general tensor $\mathcal{A}$ (defined as the minimum number of rank-one tensors into which $\mathcal{A}$ can be decomposed) is an NP-hard problem [32]. On the other hand, the model proposed by Tucker [61], as well as the related "canonical decomposition" [11] or "parallel factors" models [31], have a long history in applied data analysis [39, 40, 41, 44]. They provide exact or approximate decompositions for higher-order tensors. Recent research has focused on the relationship between these data tensor models and efforts to extend linear algebraic notions such as the SVD to multimode data tensors [44, 45, 46, 48].

A seemingly unrelated line of work has focused on matrix CUR decompositions [19, 22, 23]. As discussed in more detail in section 2.2, a matrix CUR decomposition provides a low-rank approximation of the form $A \approx \tilde{A} = CUR$, where C is a matrix consisting of a small number of columns of A, R is a matrix consisting of a small number of rows of A, and U is an appropriately defined low-dimensional encoding matrix [19]. Thus, a matrix CUR decomposition provides a dimensionally reduced low-rank approximation to the original data matrix A that is expressed in terms of a small number of actual columns and a small number of actual rows of the original data matrix, rather than, e.g., orthogonal linear combinations of those columns and rows.

In this paper, we extend a recently developed and provably accurate matrix CUR decomposition to tensor-based data sets in which there is a "distinguished" mode, and we apply it to problems in two of the three data set domains mentioned previously. When applied to hyperspectral image data, we use tensor-CUR to perform compression in order to run a classification on a more concise input, and when applied to recommendation system data, we use tensor-CUR to perform reconstruction in the absence of the full input.

By a "distinguished" mode, we mean a mode that is qualitatively different from the other modes in an application-dependent manner. The most appropriate data structure for a data set consisting of, e.g., a time-evolving internet graph or a set of hyperspectrally resolved biopsy images or user-product-product preference data for consumers depends on the application and is a matter of debate. Nevertheless, we will view such a data set as a tensor, albeit one in which one of the modes is "distinguished." For example, in these three applications, the distinguished mode would be the mode describing, respectively, the temporal evolution of the graph, the frequency or spectral variation in the images, and the users. The tensor-CUR decomposition computes an approximation to the original data tensor that is expressed as a linear combination of subtensors of the original data tensor. As we shall see, since these subtensors are actual data elements, rather than, e.g., more complex functions of data elements, in many cases they lend themselves more readily to application-specific interpretation.

2. Review of relevant linear and multilinear algebra. In this section, we provide a brief review of relevant multilinear algebra as well as recent work on matrix CUR decompositions.

2.1. Tensor-based extension of the SVD. We shall use calligraphic letters to denote higher-order or multimode tensors with $d > 2$ modes. For example, let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a d-mode tensor of size $n_1 \times n_2 \times \cdots \times n_d$ and let $N_\alpha = \prod_{i \neq \alpha} n_i$. Consider the following definitions:

- Given a tensor $\mathcal{A}$ and a particular mode $\alpha \in \{1, \ldots, d\}$, define the matrix $A_{[\alpha]} \in \mathbb{R}^{n_\alpha \times N_\alpha}$, where the columns of the matrix consist of varying the αth coordinate of $\mathcal{A}$ while leaving the rest fixed. We refer to the (usually implicit) construction of $A_{[\alpha]}$ as *matricizing* [36] or *unfolding* [44] $\mathcal{A}$ along mode α and define the *α-rank* of the tensor $\mathcal{A}$ to be the rank of the matrix $A_{[\alpha]}$.
- Given an $n_1 \times n_2 \times \cdots \times n_d$ d-mode tensor $\mathcal{A}$, a particular mode α, and any $n_\alpha \times c_\alpha$ matrix B, define the *α-mode tensor-matrix product* to be the d-mode tensor of size $n_1 \times \cdots \times n_{\alpha-1} \times c_\alpha \times n_{\alpha+1} \times \cdots \times n_d$ whose $i_1 \cdots i_d$ element is

$$(\mathcal{A} \otimes_\alpha B)_{i_1 \cdots i_d} = \sum_{i=1}^{n_\alpha} \mathcal{A}_{i_1 \cdots i_{\alpha-1} i i_{\alpha+1} \cdots i_d} B_{i i_\alpha}. \tag{1}$$

  Note that the α-mode tensor-matrix product satisfies $(\mathcal{A} \otimes_\alpha B) \otimes_{\alpha'} C = (\mathcal{A} \otimes_{\alpha'} C) \otimes_\alpha B = \mathcal{A} \otimes_\alpha B \otimes_{\alpha'} C$, assuming that the various individual products are defined.
- Given a tensor $\mathcal{A}$, let us denote the SVD of $A_{[\alpha]}$ by

$$A_{[\alpha]} = U_{[\alpha]} \Sigma_{[\alpha]} V_{[\alpha]}^T, \tag{2}$$

  where, e.g., $U_{[\alpha]}$ is an $n_\alpha \times \operatorname{rank}(A_{[\alpha]})$ matrix and $U_{[\alpha],k_\alpha}$ is an $n_\alpha \times k_\alpha$ matrix consisting of the left singular vectors corresponding to the top k_α singular values of $A_{[\alpha]}$.
- Given a d-mode tensor $\mathcal{A}$, define the (square of its) *Frobenius norm* to be

$$\|\mathcal{A}\|_F^2 = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \mathcal{A}_{i_1 \cdots i_d}^2. \tag{3}$$

- Given a tensor $\mathcal{A}$ and a particular mode α, let us refer to *slabs* as each of the n_α $d-1$-mode tensors of size $n_1 \times \cdots \times n_{\alpha-1} \times n_{\alpha+1} \times \cdots \times n_d$ constructed by fixing the αth coordinate to some particular value $i_\alpha \in \{1, \ldots, n_\alpha\}$. Similarly, let us refer to as *fibers* each of the N_α vectors (mode-one tensors) of size n_α constructed by fixing each of the other coordinates to a particular value.

Remark. See [44, 45, 36] and the references therein for a more detailed description of these tensor-related definitions. In particular, note that, although they will not be of interest to our main result, the *higher-order SVD* of $\mathcal{A}$ has been defined as the decomposition of $\mathcal{A}$ of the form $\mathcal{A} = \mathcal{S} \times_1 U_{[1]} \times_2 \cdots \times_d U_{[d]}$, where the $\operatorname{rank}(A_{[1]}) \times \cdots \times \operatorname{rank}(A_{[d]})$ tensor $\mathcal{S}$ is the so-called core tensor, and the *best rank-$(k_1, k_2, \ldots, k_d)$ approximation* to the tensor $\mathcal{A}$ has been defined as $\tilde{\mathcal{A}} = \mathcal{S} \times_1 U_{[1],k_1} \times_2 \cdots \times_d U_{[d],k_d}$. See [23] for a randomized algorithm that computes an approximation to this quantity. The algorithm of [23] is similar to the algorithms presented in this paper, except that it "unfolds" the tensor along every mode and computes an approximation to the top singular vectors of the unfolded matrix by random sampling.

Remark. Tensors are a natural generalization of matrices (see, e.g., [30] for more details) and have been studied in several fields. For example, tensors have been studied in mathematics and computer science for their algebraic properties, their ability to efficiently represent multidimensional functions, and the relationship between their properties and problems in complexity theory [30, 27, 32, 50, 8]. In addition, tensors provide a natural way to represent many large and complex data sets [44, 43, 31, 36, 46, 61, 11, 65].

Remark. The dimensionality of the linear space generated by the α-slabs is the α-rank of $\mathcal{A}$. It is worth emphasizing that computing the rank of a general tensor $\mathcal{A}$ (defined as the minimum number of rank-one tensors into which $\mathcal{A}$ can be decomposed) is an NP-hard problem, that only weak bounds are known relating the α-rank and the tensor rank, and that there do not exist definitions of tensor rank and associated tensor SVD such that the optimality properties of the matrix rank and matrix SVD are preserved [40, 33, 41, 32, 45, 38, 48, 67].

2.2. Matrix CUR decomposition. Recent work in theoretical computer science, numerical linear algebra, and statistical learning theory [19, 23, 59, 60, 7, 29, 28, 66, 22] has focused on low-rank matrix decompositions with structural properties that satisfy the following definition.

DEFINITION 1. *Let A be an $m \times n$ matrix. In addition, let C be an $m \times c$ matrix whose columns consist of a small number c of columns of the matrix A, let R be an $r \times n$ matrix whose rows consist of a small number r of rows of the original matrix A, and let U be a $c \times r$ matrix. Then $\tilde{A}$ is a* column-row-based low-rank approximation*, or a* CUR approximation*, to A if it may be explicitly written as*

$$\tilde{A} = CUR. \tag{4}$$

Several things should be noted about this definition. First, for data applications, we prefer not to provide too precise a characterization of what we mean by a "small" number of columns and/or rows, but one should think of $r, c \ll m, n$. For example, they could be constant, independent of m and n, logarithmic in the size of m and n, or simply a large constant factor less than m, n. Second, since the approximation is expressed in terms of a small number of columns and rows of the original data matrix, it will provide a low-rank approximation to the original matrix, although one with structural properties that are quite different from those provided by truncating the SVD. Third, a CUR approximation approximately expresses all of the columns of A in terms of a linear combination of a small number of "basis columns," and it does this similarly for the rows.

Finally, and most relevant for the present paper, note that a matrix CUR decomposition has structural properties that are auspicious for its use as a tool in the analysis of large data sets. For example, if the data matrix A is large and sparse but well-approximated by a low-rank matrix, then C and R (consisting of actual columns and rows) are sparse, whereas the matrices consisting of the top left and right singular vectors will not, in general, be sparse. In addition, in many applications, interpretability is important; practitioners often have an intuition about the actual columns and rows that they fail to have about linear combinations of (up to) all the columns or rows.

The following algorithmic result regarding a matrix CUR approximation was recently proven [19].

THEOREM 1. *There exists a randomized algorithm (see the* LINEARTIMECUR *algorithm of* [19]*) that takes as input an $m \times n$ matrix A and a fixed rank parameter*

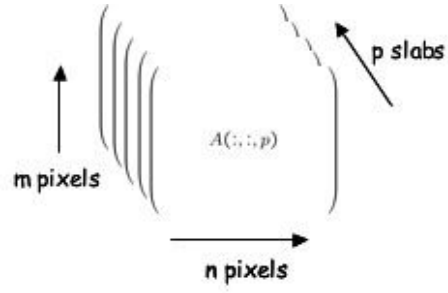


FIG. 1. *Pictorial representation of a* $(2+1)$*-data tensor.*

k and that returns as output an $m \times c$ matrix C consisting of c columns of A, an $r \times n$ matrix R consisting of r rows of A, and a $c \times r$ matrix U. The columns/rows are randomly sampled in c/r independent trials according to a judiciously chosen probability distribution depending on the Euclidean norm of the corresponding column/row. If $c = O(k \log(1/\delta)/\epsilon^4)$ and $r = O(k/\delta^2\epsilon^2)$, then

$$\|A - CUR\|_F \leq \|A - A_k\|_F + \epsilon \|A\|_F \tag{5}$$

holds with probability at least $1 - \delta$. The algorithm requires $O(m+n)$ additional time and scratch space after reading the matrix A twice from external storage.

Our two tensor-CUR algorithms are tensor-based extensions of this matrix algorithm. For more details about these results, see [17, 18, 19, 22].

3. A tensor-based extension of the matrix CUR decomposition.

3.1. A tensor-CUR decomposition for (2 + 1)-data tensors. In this subsection, for simplicity of exposition and in light of the two applications we will consider, we restrict ourselves to tensors that are subscripted by three indices, i.e., so-called 3-mode tensors.

Consider an $n_1 \times n_2 \times n_3$ tensor $\mathcal{A}$, defined as the collection of elements

$$\{\mathcal{A}_{ijk} | i = 1, \ldots, n_1; j = 1, \ldots, n_2; k = 1, \ldots, n_3\}.$$

The elements may be thought of as a data cube, i.e, a three-dimensional block such that index i runs along the vertical axis, index j runs along the horizontal axis, and index k runs along the "depth" axis. Since by assumption there is a "distinguished" mode, we are considering the special case of a $(2+1)$-*tensor*, i.e., an $n_1 \times n_2 \times n_3$ tensor in which two modes (without loss of generality, we will assume they are the first two) are similar in some application-dependent manner and the third is qualitatively different. See Figure 1 for a pictorial description of a $(2+1)$-data tensor. In this case, we refer to each of the n_3 different $n_1 \times n_2$ matrices as "slabs" and each of the $n_1 n_2$ different n_3-vectors as "fibers."

With this in mind, consider the $(2+1)$-TENSOR-CUR algorithm, described in Figure 2. This algorithm takes as input an $n_1 \times n_2 \times n_3$ tensor $\mathcal{A}$, a probability distribution $\{p_i\}_{i=1}^{n_3}$ over the slabs, a probability distribution $\{q_i\}_{i=1}^{n_1 n_2}$ over the fibers, a number c of slabs to choose, and a number r of fibers to choose. (Without loss of generality, we have assumed that the preferred mode $\alpha \in \{1, 2, 3\}$ is the third mode.) The tensor $\mathcal{A}$ is decomposed along this mode in a manner analogous to the original CUR matrix decomposition [19]. More precisely, this algorithm computes the

Input: An $n_1 \times n_2 \times n_3$ tensor $\mathcal{A}$, a probability distribution $\{p_i\}_{i=1}^{n_3}$, a probability distribution $\{q_i\}_{i=1}^{n_1 n_2}$, and positive integers c and r.
Output: An $n_1 \times n_2 \times c$ tensor $\mathcal{C}$, a $c \times r$ matrix $\mathcal{U}$, and an $r \times n_3$ matrix $\mathcal{R}$.

1. Select c slabs of $\mathcal{A}$ in c independent and identically distributed (i.i.d.) trials according to $\{p_i\}_{i=1}^{n_3}$.
   (a) Let $\mathcal{C}$ be the $n_1 \times n_2 \times c$ tensor consisting of the chosen slabs.
   (b) Let D_C be the $c \times c$ diagonal scaling matrix with $(D_C)_{tt} = \frac{1}{\sqrt{c p_{i_t}}}$ if the i_tth slab is chosen in the tth independent trial.
2. Select r fibers of $\mathcal{A}$ in r i.i.d. trials according to $\{q_i\}_{i=1}^{n_1 n_2}$.
   (a) Let $\mathcal{R}$ be the $r \times n_3$ matrix consisting of the chosen fibers.
   (b) Let D_R be the $r \times r$ diagonal scaling matrix with $(D_R)_{tt} = \frac{1}{\sqrt{r q_{j_t}}}$ if the j_tth slab is chosen in the tth independent trial.
3. Let the $r \times c$ matrix W be the matricized intersection between $\mathcal{C}$ and $\mathcal{R}$.
4. Define the $c \times r$ matrix $\mathcal{U} = D_C \left(D_R W D_C\right)^+ D_R$.

FIG. 2. *The* $(2+1)$-TENSOR-CUR *algorithm.*

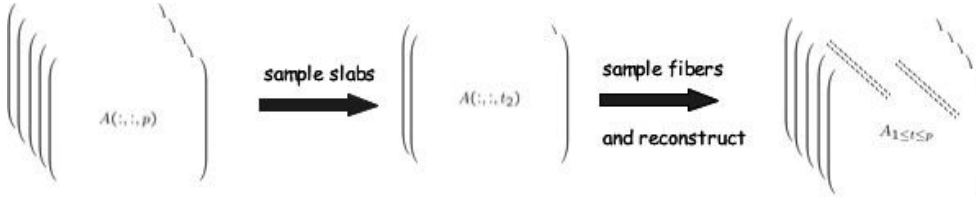


FIG. 3. *Pictorial representation of the action of the tensor-CUR decomposition.*

approximation by performing the following: first, choose c slabs (2-mode subtensors, i.e., matrices) in independent random trials and choose r fibers (1-mode subtensors, i.e., vectors) in independent random trials according to the input probability distributions; second, define the $n_1 \times n_2 \times c$ tensor $\mathcal{C}$ to consist of the c chosen slabs and also define the $r \times n_3$ matrix $\mathcal{R}$ to consist of the chosen fibers; third, let $\mathcal{U}$ be an appropriately defined and easily computed (given $\mathcal{C}$ and $\mathcal{R}$) $c \times r$ matrix.

Clearly, $\tilde{\mathcal{A}} = \mathcal{C} \otimes_3 \mathcal{U}\mathcal{R}$, where $\otimes_3$ is a tensor-matrix multiplication, is an $n_1 \times n_2 \times n_3$ tensor. Thus, by using the $(2+1)$-TENSOR-CUR algorithm, we make the following approximation:

$$\mathcal{A} \approx \tilde{\mathcal{A}} = \mathcal{C} \otimes_3 \mathcal{U}\mathcal{R}. \tag{6}$$

Thus, in particular, if $i \in 1, \ldots, n_3$ is one of the slabs that is not randomly selected, then by using the $(2+1)$-TENSOR-CUR algorithm, we make the following approximation:

$$\mathcal{A}(:,:,i) \approx \sum_{\xi \in C} \mathcal{A}(:,:,\xi) X(\xi, i), \tag{7}$$

where $\mathcal{A}(:,:,i)$ is the $n_1 \times n_2$ matrix formed from $\mathcal{A}$ by fixing the value of the third mode to be i, C is a set indicating which c indices were randomly chosen, and $X(:,i)$ is a vector consisting of the ith column of the matrix $\mathcal{U}\mathcal{R}$.

See Figure 3 for a pictorial description of the action of the algorithm and this approximation. In particular, note that a small number of slabs are sampled, and every other slab is approximately reconstructed using the information in those slabs as

a basis along with the information in a small number of fibers (depicted as the dashed lines). The extent to which (6) or (7) is a good approximation has to do with the selection of slabs and fibers. In sections 4 and 5, we show that (6) holds empirically for our two applications if the slabs and fibers are chosen uniformly and/or nonuniformly with probabilities that depend on the Frobenius norms of slabs and Euclidean norms of fibers, respectively. See the proof of Theorem 2 in section 3.2 and also [17, 18, 19] for a discussion of the algorithmic justification for this sampling.

We emphasize that, as with the matrix CUR decomposition, when this tensor-CUR decomposition is applied to data, there is a natural interpretation in terms of underlying data elements. For our imaging application, a "slab" corresponds to an image at a given frequency step and a "fiber" corresponds to a time- or frequency-resolved pixel. Similarly, for our recommendation system application, a "slab" corresponds to a product-product preference matrix for a single user and a "fiber" corresponds to preference information from every user about a single product-product pair.

3.2. A general tensor-CUR decomposition for very large data tensors. In this subsection, to provide a theoretical justification for the tensor-CUR decomposition of section 3.1, we present our main algorithmic result. Our main algorithmic result is a generalization of the $(2+1)$-Tensor-CUR algorithm and an associated provable quality-of-approximation bound for the Frobenius norm of the error tensor $\mathcal{A} - \mathcal{C} \otimes_3 \mathcal{U}\mathcal{R}$.

The Tensor-CUR algorithm, described in Figure 4, takes as input a d-mode tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, a "distinguished" mode $\alpha \in \{1, \ldots, d\}$, a rank parameter k_α, an error parameter $\epsilon > 0$, and a failure probability $\delta \in (0, 1)$. The algorithm returns as output three carefully constructed subtensors that, when multiplied together, are an approximation $\tilde{\mathcal{A}}$ to $\mathcal{A}$. Both the number of slabs c_α and the number of fibers r_α that are randomly sampled depend on the rank parameter k_α, an error parameter ϵ, and a failure probability δ. The subtensors $\mathcal{C}$ and $\mathcal{R}$ are chosen by sampling according to a carefully constructed nonuniform probability distribution. In order to obtain the provable quality-of-approximation bounds of Theorem 2, the probability distribution depends on the Frobenius norms of the slabs and the Euclidean norms of the fibers, respectively. Intuitively, this biases the random sampling toward the subtensors that are of most interest; see [17, 18, 19] for details.

In more detail, the approximation $\tilde{\mathcal{A}}$ is computed by performing the following: first, form (implicitly) each of the n_α subtensors (slabs of mode $d-1$) defined by fixing $i \in \{1, \ldots, n_\alpha\}$, and also form (implicitly) each of the $N_\alpha = \prod_{i \neq \alpha} n_i$ subtensors (fibers of mode 1, i.e., vectors) defined by fixing a value for each of the modes $i \neq \alpha$; second, construct nonuniform probability distributions with which to sample the slabs and the fibers; third, choose c_α of the $d-1$-mode slabs in independent random trials, and also choose r_α of the 1-mode fibers in independent random trials; fourth, define the tensor $\mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_{\alpha-1} \times c_\alpha \times n_{\alpha+1} \times \cdots \times n_d}$ to consist of the c_α chosen $d-1$-mode slabs, and also define the tensor $\mathcal{R} \in \mathbb{R}^{r_\alpha \times n_\alpha}$ to consist of the r_α chosen 1-mode fibers; and finally, let $\mathcal{U} \in \mathbb{R}^{c_\alpha \times r_\alpha}$ be an appropriately defined and easily computed (given $\mathcal{C}$ and $\mathcal{R}$) tensor of mode 2 (i.e., matrix). Then we may define

$$\tilde{\mathcal{A}} = \mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R}, \tag{8}$$

where $\mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R}$ is the α-mode tensor-matrix product between $\mathcal{C}$ and $\mathcal{U}\mathcal{R}$, to be an $n_1 \times \cdots \times n_{\alpha-1} \times n_\alpha \times n_{\alpha+1} \times \cdots \times n_d$ tensor that is an approximation to the original tensor $\mathcal{A}$. (The awkward form of $\mathcal{U}$ is currently necessary for our provable results. Nevertheless, $\mathcal{U}$ is a subspace perturbation of the Moore–Penrose generalized inverse

Input: An $n_1 \times n_2 \times \cdots \times n_d$ tensor $\mathcal{A}$, a mode $\alpha \in \{1, \ldots, d\}$, a rank parameter k_α, an error parameter $\epsilon > 0$, and a failure probability $\delta \in (0, 1)$.
Output: An $n_1 \times \cdots \times n_{\alpha-1} \times c_\alpha \times n_{\alpha+1} \times \cdots \times n_d$ tensor $\mathcal{C}$, a $c_\alpha \times r_\alpha$ matrix $\mathcal{U}$, and an $r_\alpha \times n_\alpha$ matrix $\mathcal{R}$.

1. Let $c_\alpha = 4k_\alpha\big(1 + \sqrt{8\log(2/\delta)}\big)^2/\epsilon^4$, $r_\alpha = 4k_\alpha/\delta^2\epsilon^2$, and $N_\alpha = \prod_{i \neq \alpha} n_i$.
2. Define $\{p_i\}_{i=1}^{n_\alpha}$ to be $p_i = \frac{\left|(A_\alpha)^{(i)}\right|^2}{\|\mathcal{A}\|_F^2}$.
3. Define $\{q_j\}_{j=1}^{N_\alpha}$ to be $q_j = \frac{\left|(A_\alpha)_{(j)}\right|^2}{\|\mathcal{A}\|_F^2}$.
4. Select c_α slabs of $\mathcal{A}$ in c_α i.i.d. trials according to the probability distribution $\{p_i\}_{i=1}^{n_\alpha}$.
   (a) Let $\mathcal{C}$ be the $n_1 \times \cdots \times n_{\alpha-1} \times c_\alpha \times n_{\alpha+1} \times \cdots \times n_d$ tensor consisting of the chosen slabs.
   (b) Let D_C be the $c_\alpha \times c_\alpha$ diagonal scaling matrix with $(D_C)_{tt} = \frac{1}{\sqrt{c p_{i_t}}}$ if the i_tth slab is chosen in the tth independent trial.
5. Select r_α fibers of $\mathcal{A}$ in r_α i.i.d. trials according to the probability distribution $\{q_i\}_{i=1}^{N_\alpha}$.
   (a) Let $\mathcal{R}$ be the $r_\alpha \times n_\alpha$ matrix consisting of the chosen fibers (from all the slabs).
   (b) Let Ψ be the $r_\alpha \times c_\alpha$ matrix consisting of the chosen fibers (from the chosen slabs).
   (c) Let D_R be the $r_\alpha \times r_\alpha$ diagonal scaling matrix with $(D_R)_{tt} = \frac{1}{\sqrt{r q_{j_t}}}$ if the j_tth slab is chosen in the tth independent trial.
6. Let Φ be the best rank-k approximation to the Moore–Penrose generalized inverse of $(\mathcal{C} \otimes_\alpha D_C)_{[\alpha]}^T (\mathcal{C} \otimes_\alpha D_C)_{[\alpha]}$.
7. Define the $c_\alpha \times r_\alpha$ matrix $\mathcal{U} = \Phi\,(D_R\Psi)^T$.

FIG. 4. *The* TENSOR-CUR *algorithm.*

of the matricized intersection between $\mathcal{C}$ and $\mathcal{R}$. Thus, for the $(2+1)$-TENSOR-CUR algorithm and for the applications described in sections 4 and 5, we have taken it to be exactly this quantity.)

Our main quality-of-approximation bound for the TENSOR-CUR algorithm is given by the following theorem, in which we bound the Frobenius norm of the error tensor $\tilde{\mathcal{E}} = \mathcal{A} - \tilde{\mathcal{A}}$.

THEOREM 2. *Let $\mathcal{A}$ be an $n_1 \times n_2 \times \cdots \times n_d$ tensor, and let $\alpha \in \{1, \ldots, d\}$ be a particular mode, k_α be a rank parameter, $\epsilon > 0$ be an error parameter, and $\delta \in (0, 1)$ be a failure probability. Construct a tensor-CUR approximate decomposition to $\mathcal{A}$ with the output of the* TENSOR-CUR *algorithm. Then, with probability at least $1 - \delta$,*

$$\|\mathcal{A} - \mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R}\|_F \leq \left\|A_{[\alpha]} - \left(A_{[\alpha]}\right)_{k_\alpha}\right\|_F + \epsilon \|\mathcal{A}\|_F . \tag{9}$$

Proof. Since "unfolding" $\mathcal{A}$ along any mode does not change the value of its Frobenius norm (as it is simply a reordering of indices in a summation), it follows that

$$\|\mathcal{A} - \mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R}\|_F = \left\|A_{[\alpha]} - (\mathcal{C} \otimes_\alpha \mathcal{U}\mathcal{R})_{[\alpha]}\right\|_F . \tag{10}$$

Note that the Frobenius norm on the left-hand side of (10) is a tensor norm and that the Frobenius norm on the right-hand side of (10) is a matrix norm. Due to the form

of the sampling probabilities used in the TENSOR-CUR algorithm, it is this latter quantity that Theorem 5 of [19] bounds. By applying this result [19], the theorem follows. □

3.3. Remarks on tensor-CUR decompositions and data applications.

Remark. In (9), the $\left\|A_{[\alpha]} - (A_{[\alpha]})_{k_\alpha}\right\|_F$ term is a measure of the extent to which the "unfolded" matrix $A_{[\alpha]}$ is not well-approximated by a rank-k_α matrix, and the $\epsilon \|\mathcal{A}\|_F$ term is a measure of the loss in approximation quality due to the choice of slabs and fibers (rather than, e.g., the top k_α eigenslabs and eigenfibers along the α mode). This latter measure is of the form of an arbitrary (but fixed) precision, scaled by a measure of the size of the tensor $\mathcal{A}$.

Remark. The values for c_α and r_α in general differ, as they do with matrix CUR decompositions. Although this is an artifact of the proof techniques [19], this allows for greater flexibility in data applications. For example, if the noise properties of the slabs and fibers differ, then one may wish to oversample the slabs or fibers in different ways.

Remark. The choice for slabs and fibers in the TENSOR-CUR algorithm takes advantage only of linear and not multilinear structure in the data tensors. Equivalently, the algorithm reduces to the corresponding matrix algorithm. It is an open problem whether one can choose slabs and/or fibers to preserve some nontrivial multilinear tensor structure in the original tensor $\mathcal{A}$.

Remark. A crucial decision in applying these techniques to data will be the proper choice (if any) of the preferred mode α. This depends on the application area from which the data are drawn. The theorems will be true but uninteresting if this choice is not made carefully.

Remark. Assume, for simplicity, that the tensor $\mathcal{A}$ is stored externally, and assume that $k_i = O(1)$ and that $n_i = n$ for every $i = 1, \ldots, d$. Then the matrices $C_{[i]}$ each occupy only $O(n)$ additional scratch space. In general, $O(n^{d-1})$ additional scratch space will be needed to compute the probabilities of the form used by the TENSOR-CUR algorithm, and this will be comparable to the overall memory requirements if d is large. On the other hand, if the uniform probabilities are approximately optimal for each of the d nodes, then only $O(n)$ additional scratch space and computation time are needed, resulting in a substantial scratch memory and time savings. See [17] for additional discussion of resource issues within the framework of the pass-efficient model of data streaming computation.

Remark. Although sampling with respect to the proper probability distribution is critical for our provable results, one might expect that in many cases the slabs and/or fibers will all be approximately the same length due to the manner in which the data are generated, in which case uniform sampling may be successfully employed. This was seen to be the case for an application of the CUR algorithm of [19] to kernel-based learning [21, 22, 66].

Remark. Alternatively, one might expect that in many cases the data are generated in such a way that information about the Frobenius norm of each of the slabs and/or fibers is easily computed at the data generation step. For example, in the case of a $(2+1)$-imaging application, the Frobenius norm of a slab corresponds to the total absorption at one time step or frequency value. In this case, these approximations to the probabilities could be used in the TENSOR-CUR algorithm.

Remark. Although $c_\alpha = 4k_\alpha\left(1+\sqrt{8\log(2/\delta)}\right)^2/\epsilon^4$ slabs and $r_\alpha = 4k_\alpha/\delta^2\epsilon^2$ fibers suffice to prove the claims of Theorem 2, they can be rather large for even moderate values of k_α, δ, and ϵ. In the applications we consider, choosing many fewer slabs and

fibers suffices, e.g., on the order of tens or hundreds; see sections 4 and 5 for more detail.

4. Application to hyperspectral image data. In hyperspectral imagery, an object or scene is imaged at a large number of contiguous wavelengths [51]. Although hyperspectral imagery originated in astronomy and geosensing, it has been employed more recently in numerous other application areas, including agriculture, manufacturing, forensics, and medicine. In many of these applications, target resolution is limited by available spatial resolution. By considering the spectral variation of light intensity, one obtains rich information about the object or scene being imaged that complements traditional spatial information. One also obtains very large data sets that may be represented as a tensor and that contain much redundancy. For example, if a single scene is imaged at 128 frequency bands, where at each frequency a 495×656 image is generated, then the data cube generated for this single object consists of 40 million values and may be represented by a $495 \times 656 \times 128$ tensor $\mathcal{A}$, where $\mathcal{A}_{ijk}$ represents the absorbed or transmitted light intensity at pixel ij at physical frequency k.

In this section, we describe an application of the tensor-CUR decomposition to a problem in hyperspectral medical image analysis. In particular, the tensor-CUR decomposition is used to *compress* the data, and we show that tissue segmentation and nuclei classification quality are not substantially reduced even after substantial data compression. In more detail, in section 4.1, we describe the data and its generation. Then, in section 4.2, we describe the reconstruction of the full data from a small sample of slabs and fibers. In section 4.3, we describe the classification task of tissue segmentation, i.e., classifying the pixels in a single image into different tissue types, as a function of how heavily we downsample on the slabs and fibers. This task is of intermediate interest, since nuclei are the most discriminative structures in the final classification task of interest. Finally, in section 4.4, we describe the classification of data cubes into, e.g., normal and malignant, as a function of downsampling on the slabs and fibers.

4.1. Description of data and data generation. The application of hyperspectral imaging to medicine, and pathology in particular, while not new, is becoming more widespread and powerful. A variety of proprietary spectral splitting devices, including prisms and mirrors [64], interferometers [25, 55], variable interference filter-based monochromometers [53], and tuned liquid crystals [47], mounted on microscopes in combination with CCD cameras and computers, have been used to discriminate among cell types, tissue patterns, and endogenous and exogenous pigments [47]. Although the increasing power of these methods holds the promise for developing automatic diagnostics, the increased volume and formal dimensionality of the data make the development of more efficient algorithms necessary in order to extract statistically useful and reliable information about the data.

The prototype-tuned light source used to generate the data we studied (Plain Sight Systems, Inc.; see [16] for details) can generate a large number of combinations of light frequencies, ranging from about 440 nm to about 700 nm, with a wavelength resolution of up to approximately 6 nm. The light modulated by the prototype is shone via a fiber optic cable directed in a Nikon Biophot microscope and transilluminates hematoxylin and eosin (H & E) stained microarray tissue sections of normal, benign (adenoma), and malignant carcinoma colon biopsies. Hyperspectral photomicrographs, collected in random order at 400X magnification, are obtained with a CCD camera (Sensovation) from 59 different patient biopsies (20 normal, 19 benign ade-

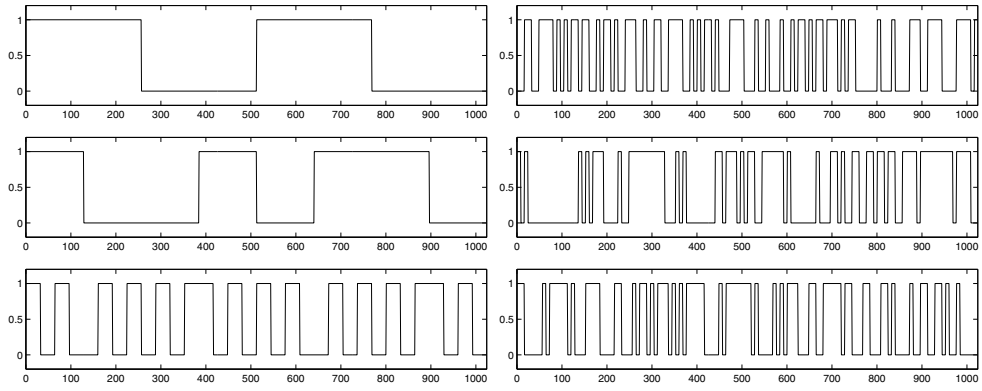


FIG. 5. *Examples of Hadamard patterns (left) and randomized Hadamard patterns (right). The latter are used at the data generation step to improve the signal-to-noise ratio; see the text for details.*

noma, and 20 malignant carcinoma), mounted as a microarray on a single glass slide [14, 2, 3, 5]. From these, 59 hyperspectral grayscale images at 400X magnification are derived. The biopsies are collected randomly on the slide across and within the different groups of biopsies in order to avoid any biases due to instrumentation, e.g., due to temperature or time of collection. This data was collected by G. L. Davis, M.D., as discussed in [51].

Each measurement yields a data cube, which is a set $\{I_i\}_{i=1\ldots128}$ of 128 images, each of which is 495 by 656 pixels in size. (That is, there is one data cube for each of the 59 biopsies.) The intensity of the pixel $I_i(x,y)$ is (ideally) the transmitted light at location (x,y) when the ith light pattern ψ_i shone through the sample. The data is collected by using randomized Hadamard patterns in order to maximize the signal-to-noise ratio. The noise in the measurement of the hyperspectral image can be modeled as independent of the intensity of light shown through the sample. The signal-to-noise ratio of the measurement of each I_i, for a fixed integration time for the measurement of I_i, is maximized when the amount of modulated light shone through the sample is maximized. The instrument allows us to shine patterns $\{\psi_i\}_{i=1,\ldots,S} = \{\psi_i(\nu) = \sum_{j=1}^{N} \epsilon_{ij}\delta_j(\nu)\}$, where $(\epsilon)_{ij}$ is an S by N matrix with entries in $\{0,1\}$, and $(\delta)_j$, an S-dimensional vector, represents (ideally) a Dirac δ-function at physical frequency $\nu_j \sim (700-440)j/N+440$. In our experiment, we set $N = 256$ (the instrument would allow up to $N = 1024$) and $S = 128$. Ideally, $I_i(x,y)$, $i = 1,\ldots,S$, is the value of the inner product (in the frequency variable ν)

$$I_i(x,y) = \langle f(x,y,\nu), \psi_i(\nu)\rangle_\nu = \sum_j f(x,y,\nu_j)\psi_i(\nu_j)\,,$$

where $f(x,y,\nu)$ is the transmittance of the sample at location (x,y) and frequency ν. The choice of the patterns ψ_i is crucial in determining the signal-to-noise ratio of the measurements for a fixed integration time and total intensity of the light source: we use the idea of *multiplexing* and shine a sequence of *randomized Hadamard patterns* $\{\psi_i^H\}_{i=1,\ldots,N}$, obtained from standard Hadamard patterns by randomly shuffling the frequency axis. See Figure 5 for examples of Hadamard and randomized Hadamard patterns.

Thus, each data cube consists of 128 images, each 495 by 656 pixels in size (for a

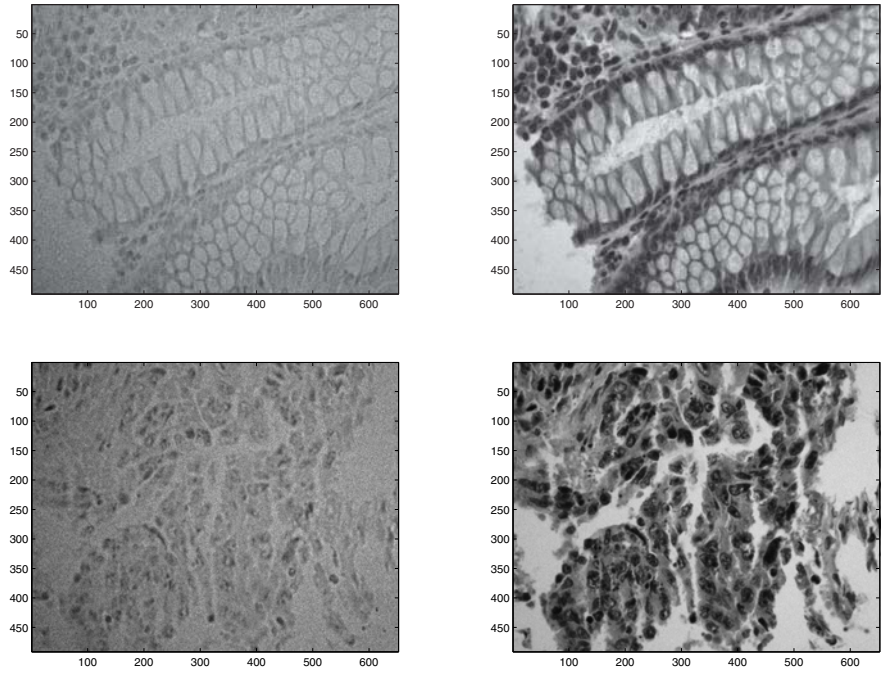


FIG. 6. *Two different spectral slices, i.e., two different images each at a single frequency, from a hyperspectral data cube derived from a normal sample (top) and from a hyperspectral data cube derived from a very malignant sample (bottom).*

total of about 40 million pixels), measuring the modulated light transmitted through the sample. We view this as a $495 \times 656 \times 128$ 3-mode tensor $\mathcal{A}$, where the entry $\mathcal{A}_{ijk}$ is proportional to the light with spectral modulation k transmitted at location (i, j). Each biopsy contains either normal, benign (adenoma), or malignant (cancerous) tissue and is labeled by G. L. Davis, M.D., pathologist. Various algorithms have previously been shown to find and classify automatically normal, abnormal, and malignant small portions of each biopsy [14, 15, 51] using the complete data cube. As we describe in more detail in the next three subsections, we couple the tensor-CUR decomposition described in section 3.1 with ideas from [14, 15, 51] in order to speed up computations, denoise, compress, and preprocess the data, and we show that this causes only a small loss of performance of these algorithms.

In order to gain a feel for the data, consider Figures 6 and 7. Figure 6 illustrates two of the 128 images, i.e., two hyperspectral images at two distinct frequencies, in a normal sample and in a very malignant sample. Similarly, Figure 7 illustrates a typical frequency-resolved pixel in both a normal and a malignant nucleus as well as a single spectrum in the malignant sample and the spectrum averaged over every one of the ca. 324,000 frequency-resolved pixels in the malignant data cube. Note that both successive images and pixels from different spatial regions are strongly correlated with one another.

In this imaging application, the tensor $\mathcal{C}$ in the tensor-CUR decomposition consists of a small number of dictionary or basis images (which are actual and not eigenimages) with respect to which the remaining images are expressed. Similarly, the

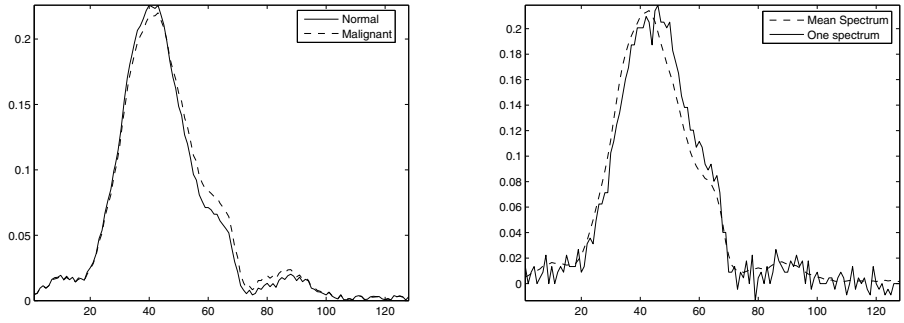


FIG. 7. *Left: Average normalized nuclei spectrum from a normal and a malignant sample. Right: Average normalized spectrum and a single typical spectrum in one hyperspectral data cube. The vertical axis represents normalized energy per frequency in the spectra, and the horizontal axis is the slab index.*

matrix $\mathcal{R}$ consists of the spectral variation of a small number of dictionary or basis pixels with respect to which spectral variation of the remaining pixels is expressed. In the next three subsections, we will see that the tensor-CUR decomposition can be applied to this hyperspectral image data in order to compress the data and to perform two classification tasks of interest on the data. That is, the tensor-CUR algorithm will downsample slabs $\mathcal{A}(:,:,\nu_i)$ by sampling a set of images at certain randomly chosen wavelengths $\{\nu_i\}_{i=1}^{128}$ and fibers $\mathcal{A}(x_i, y_i, :)$ by sampling spectra at certain randomly chosen locations $\{(x_i, y_i)\}$. Slabs will be chosen randomly with a probability proportional to the average normalized spectrum of Figure 7, i.e., with probability proportional to $||\mathcal{A}(:,:,\nu)||_F$, and fibers will be chosen uniformly at random. The data-dependent motivation for this is that the intensity of transmitted light captures a meaningful notion of information as a function of varying frequency but not as a function of varying spatial coordinates due to the particular staining technology.

4.2. Reconstruction of hyperspectral data. For each slab we did not randomly sample, we use the tensor-CUR decomposition to reconstruct that slab in the basis provided by the sampled slabs, and we do so using only a small number of pixels in that slab. In Figure 8, we present a representative example of the reconstruction of two spectral slices from a normal biopsy and two spectral slices from a malignant biopsy. The redundancy in the data is evident by the quality of the reconstruction under very heavy downsampling. For example, it suffices to judiciously choose as few as 8 or even 2 of the original 128 slabs, and to reconstruct the remaining slabs, it suffices to choose ca. 1000 (or fewer) of the original ca. 324,000 fibers.

In Figure 9, we present the approximation error as a function of downsampling to different numbers of slabs and then to different numbers of fibers. As expected, as the number of sampled slabs and fibers increases, the approximation error decreases. The approximation error is very small in the middle range of the frequencies, where the energy per frequency is larger, and hence the sampling probability is larger. Thus, due to the form of the slab sampling probabilities, slabs between ca. 30 and ca. 60 tend to be reproduced much better than those toward the tails of the spectrum. Slabs below ca. 20 and above ca. 70 tend to have a lower signal-to-noise ratio and are less important for the problem of approximate data reconstruction (but not necessarily for other problems). Sampling more than 1200 fibers does not lead to significant

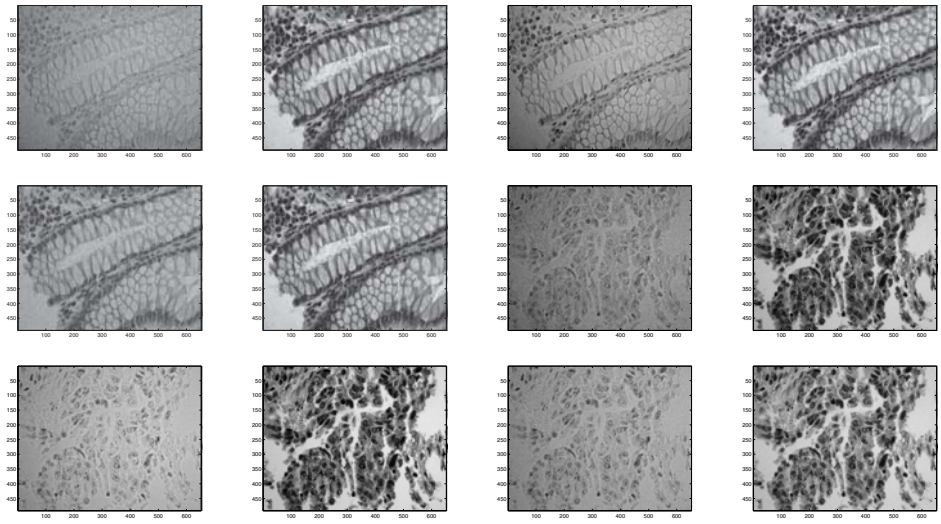


FIG. 8. *Typical reconstruction of the hyperspectral data cubes as a function of sampling. Shown in this figure are two different spectral slabs from a normal biopsy and two from a malignant biopsy, each reconstructed under three different compression ratios. In particular, the three figures in the first column are from slab number* 30 *(out of* 128*) from a normal sample; the second column is from slab number* 60 *from the same normal sample; and the third and fourth columns are slabs number* 30 *and* 60*, respectively, from another biopsy that is malignant. Presented are the original data (in the top row), the data when it is compressed with* 8 *slabs and* 1200 *fibers (in the middle row), and even more compressed data with only* 2 *slabs and* 1200 *fibers (in the bottom row).*

improvements (unless several tens of thousands fibers are sampled).

At this point, we observe that the spectra reconstructed after compression are far less noisy than the original spectra. More precisely, a close examination of images such as those presented in Figure 8 reveals a subtle interplay between sampling-induced error and denoising due to the low dimensionality of the sample. This has a denoising and a regularization effect on the spectra, and we can interpret the low-dimensional projection achieved by compression as a denoising mechanism, tuned to each data cube. Note that by giving our tensor-CUR algorithm the flexibility to sample different numbers of slabs and fibers, we can, e.g., sample slabs to a level appropriate for structure identification and sample more fibers for denoising purposes.

4.3. Tissue-type segmentation. In medical applications, one is interested in the classification of an entire data cube, i.e., a medical sample, as normal or malignant. Biological reasons suggest that nuclei are the most discriminative structures for this task. Thus, as an intermediate step, one is interested in classifying the pixels in a single data cube into different tissue types, e.g., nuclei, cytoplasm, or lamina propria, based on the spectral response ("fiber") associated with each pixel. For each of the 59 images, we use the algorithm described in [14, 15, 51] for segmenting the pixels in the image into three sets of regions corresponding to different tissue types. This algorithm is based on the local discriminant basis (LDB) algorithm [13, 56, 57] to find features that best discriminate among the different classes and a nearest neighbor classifier in a discriminant projection found by LDB. Note that for the normal versus malignant classification task of the next subsection (in which we classify entire data cubes), we have access to a label (assumed correct) provided by a pathologist [51],

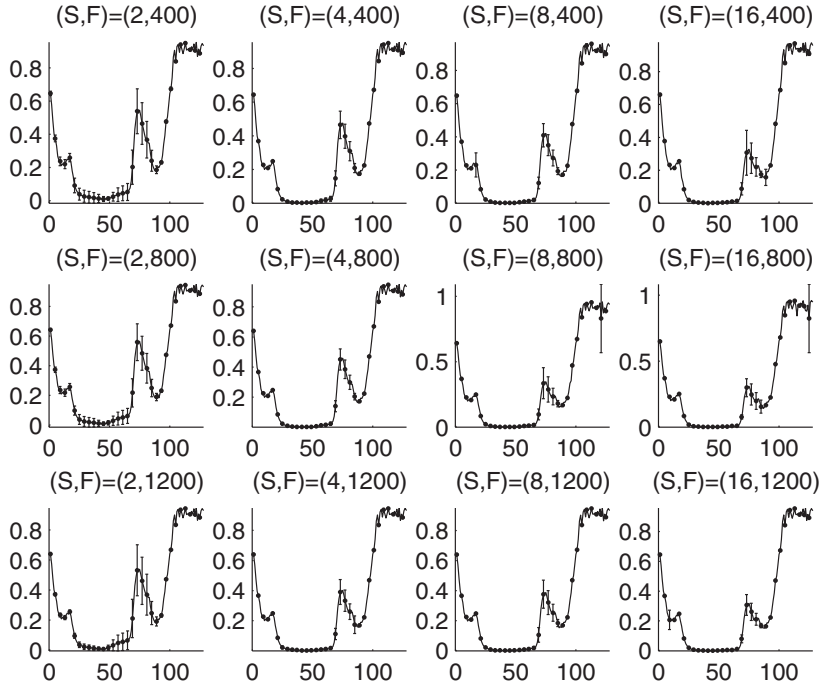


FIG. 9. *Reconstruction error. The caption indicates how many slabs (S) and fibers (F) were sampled. The vertical axis is the relative reconstruction error (for the Frobenius norm). The horizontal axis is the slab index. Average and standard deviation are over* 4 *slab draws and* 3 *fiber draws.*

while no such ground truth is available for this tissue classification in this section (in which we classify pixels in each image).

In Figure 10, we present typical results for running this tissue classification algorithm on two data cubes (one normal and one malignant) with increasing compression ratios. We see that the tissue classification is affected in two different ways. When we sample 16 slabs, the tissue classification, at least qualitatively speaking, improves by becoming less noisy and by generating fewer misclassification errors. See, e.g., the isolated red pixels, which correspond to nuclei, in the images in the leftmost column of Figure 10. As the compression ratio increases further, we observe a slight decreased performance in the tissue classification algorithm. As with the reconstruction problem, in both cases there is little quality loss until the number of fiber samples is less than ca. 1000. In addition, as before, a careful analysis reveals a complex interplay between sampling-induced information loss and sampling-induced denoising. Unfortunately, it is not possible for us to quantify these results, since this would require an individual to mark, by hand and with high precision, the correct tissue segmentation.

4.4. Classification of nuclei and data cubes. If the nuclei identified by the tissue classification described in section 4.3 are then used to classify data cubes, the results can be compared with the true value (assigned by the pathologist). For each nucleus, we consider the mean spectrum, and we use partial least squares (PLS) to build a linear classifier to classify this spectrum. We consider the following two

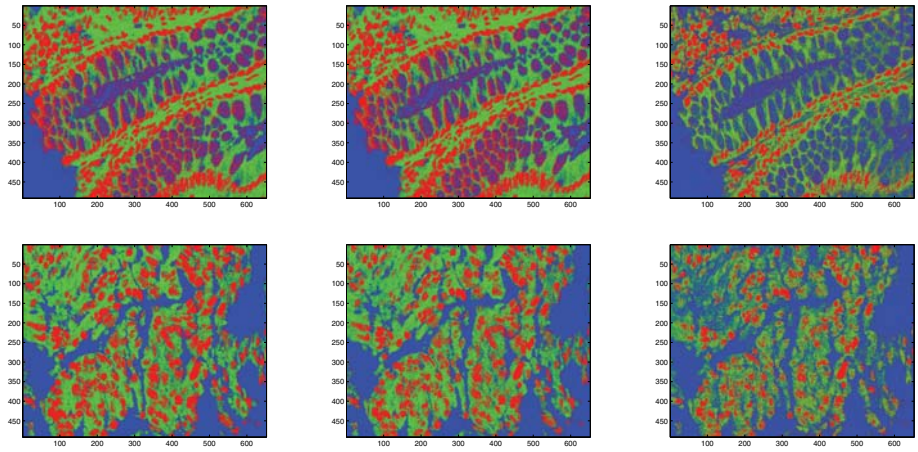


FIG. 10. *Segmentation into three tissue types in a normal biopsy (top row) and a malignant biopsy (bottom row): red for nuclei (the only class that we are interested in for the next classification task), green for cytoplasm, and blue for lamina propria and other regions. From left to right: classification on original data; on compressed data (*16 *slabs and* 1200 *fibers); and on compressed data (*8 *slabs and* 1200 *fibers).*

classification tasks: classify as normal or malignant; and classify as normal, abnormal, or malignant. In addition, we use two cross-validation procedures, described below, for each classification task. See [14, 15, 51] for more details on these procedures.

We define the patches we want to classify as follows. A patch is a subset of a data cube of the form $Q^l_{x_0,y_0} \times S$, where $Q^l_{x_0,y_0}$ is a square of side l pixels long, centered at (x_0, y_0), and S denotes the complete spectral range. A patch is *admissible* if it contains at least $\frac{8}{10}l^2$ nuclei pixels. From now on, we will consider each patch simply as a collection of the nuclei spectra it contains and hence as a cloud in $\mathbb{R}^{128}$. For the results reported here, we have chosen and fixed $l = 64$, which provides a size that roughly corresponds to the size of a single nucleus. The set of $l \times l$ patches we consider consists of 3298 patches chosen by the algorithm by randomly picking a square in the slide and checking if it is admissible. About 60 patches per slide are collected. We denote by $\{N_{i,k}\}_{k \in K_i}$ the set of nuclei spectra in the ith patch P_i.

For each admissible patch P_i collected, we compute the mean of the nuclei spectra $\{N_{i,k}\}_k$, and we normalize it to unit energy. We denote this set of normalized average nuclei spectra by $\mathcal{N}$. (Therefore, $|\mathcal{N}| = 3298$, as above.) The label (e.g., normal or abnormal) attached to the patch is transferred to the corresponding mean nucleus spectrum. We used PLS, keeping $k = 15$ top vectors, and we ran 50 rounds of 25-fold cross-validation to avoid overfitting. We run this cross-validation in two different ways:

- (Weak CV) Extract a random training subset of size $\frac{3}{4}|\mathcal{N}|$ and predict on the remaining subset of size $\frac{1}{4}|\mathcal{N}|$.
- (Strong CV) Extract a random subset of biopsies, of size $\frac{3}{4}$#biopsies, train the algorithm on the corresponding normalized average nuclei in $\mathcal{N}$ extracted from those biopsies, and test the algorithm on the remaining subset of $\mathcal{N}$, corresponding to averaged normalized nuclei extracted from the remaining biopsies.

Thus, in each case, the training and testing sets are subsets of biopsies. Note that the

TABLE 1

Confusion matrix of predictions of normal and malignant nuclei (patches of size 64 *by* 64 *with averaged* 25*-fold cross-validated error) using average (weak CV) error. TN, TM stand for true normal and true malignant, and PN, PM stand for the corresponding predictions. From left to right, the number of random slabs sampled is* $128(\text{all}), 16, 8, 2$.

	PN	PM
TN	90%	10%
TM	10%	90%

	PN	PM
TN	100%	0%
TM	0%	100%

	PN	PM
TN	100%	0%
TM	0%	100%

	PN	PM
TN	100%	0%
TM	0%	100%

TABLE 2

Confusion matrix of predictions of normal, benign (adenoma), and malignant nuclei patches, as in Table 1*, but errors corresponding to (strong CV).*

	PN	PM
TN	79%	21%
TM	26%	74%

	PN	PM
TN	77%	23%
TM	30%	70%

	PN	PM
TN	79%	22%
TM	29%	71%

	PN	PM
TN	68%	32%
TM	33%	67%

first cross-validation is weaker. Since we expect correlations between (nuclei) spectra in the same data cube, and since in (weak CV) the training set contains, with high probability, nuclei spectra from all the biopsies, training and testing sets cannot be assumed to be completely independent. Most of this lack of independence, we think, is due to normalization issues, sample preparation, lighting, and other data collection conditions, which exhibit variations across biopsies. Since we can consider different biopsies as being independent samples as they were collected in random order and independently of the type (e.g., normal, abnormal, or malignant), the second cross-validation is stronger.

We are interested in measuring any change of performance of the classification algorithm as a function of the compression ratio. The confusion matrices of the classifiers obtained are summarized in Tables 1, 2, 3, and 4 for classifiers of patches of size $l = 64$. These confusion matrices are averages over the performance on the testing set in several cross-validation runs. For the full data, the two-class discrimination between normal and carcinoma nuclei correctly identifies 79% of normal and 74% of malignant nuclei. The three-class discrimination among normal, abnormal (adenoma), and carcinoma nuclei is much more challenging (independently of compression), with identification rates of 33%, 73%, and 40% for normal, abnormal, and carcinoma samples, respectively. We study how this performance changes under compression of the data cubes. As can be seen, in general, high quality results are obtained using samples of 16 and 8 slabs, but quality degrades if only 2 slabs are used. Also, note that the algorithm performs more poorly (about 25% error in the discrimination of the 3 classes of biopsies) on completely new biopsies. This is related to normalization of the data, due both to the process of staining and to the instrument calibration and data collection. Current research is addressing these issues.

In Tables 1 and 2, we classify normal and malignant, and then we run the same classifier on data cubes compressed at different compression ratios; we also show the difference between weak and strong cross-validation. Observe that the performance of the algorithm is very good across compression ratios, except for a significant decrease of performance for a very high compression ratio (sampling of only 2 slabs!). We interpret this as a balancing effect between the possible loss of information due to compression and the denoising and regularization effect due to the dimensionality reduction.

In Tables 3 and 4, we classify normal, abnormal, and malignant, and again we

TABLE 3

Confusion matrix of predictions of normal, benign (adenoma), and malignant nuclei (patches of size 64 *by* 64 *with averaged* 25*-fold cross-validated error) using average (weak CV) error. TN, TB, TM stand for true normal, true benign (adenoma), and true malignant, and PN, PB, PM stand for the corresponding predictions. From left to right, the number of random slabs sampled is* $128(\text{all}), 16, 8, 2$.

	PN	PB	PM
TN	45%	51%	4%
TB	17%	76%	7%
TM	4%	36%	60%

	PN	PB	PM
TN	96%	4%	0%
TB	1%	98%	0%
TM	0%	3%	97%

	PN	PB	PM
TN	99%	1%	0%
TB	0%	100%	0%
TM	0%	1%	99%

	PN	PB	PM
TN	100%	0%	0%
TB	0%	100%	0%
TM	0%	0%	100%

TABLE 4

Confusion matrix of predictions of normal, benign (adenoma), and malignant nuclei patches, as in Table 3*, but with (strong CV).*

	PN	PB	PM
TN	33%	61%	6%
TB	22%	73%	5%
TM	9%	51%	40%

	PN	PB	PM
TN	42%	24%	34%
TB	31%	36%	33%
TM	23%	28%	49%

	PN	PB	PM
TN	30%	53%	17%
TB	26%	61%	13%
TM	7%	48%	51%

	PN	PB	PM
TN	30%	45%	25%
TB	29%	53%	16%
TM	12%	35%	53%

run the same classifier on data cubes compressed at different compression ratios; we also show the difference between weak and strong cross-validation. Here we observe a interesting phenomenon: under (weak CV), the algorithm performs much more poorly on the original data than on the compressed data. Hence the compression has a regularization effect that greatly helps the learning phase. This advantage is partly lost when we consider the (strong CV). Of course, the three-class problem is expected to be much harder than the two-class problem, not only because, from a machine-learning perspective, multiclass problems are harder but also because the abnormal samples are often quite similar to normal samples, and even in the field of pathology, the differences are qualitative and often not large.

5. Application to recommendation system analysis. In recommendation system analysis, one is typically interested in making purchase recommendations to a user at an electronic commerce web site. Collaborative methods (as opposed to content-based or hybrid) involve recommending to the user items that people with similar tastes or preferences liked in the past. Probably the most well-known example of a collaborative filtering system is that of Amazon.com, which is based on rules of the form "users who are interested in item X are also likely to be interested in item Y" [49]. Many collaborative filtering algorithms represent a user as an n-dimensional vector, where n is the number of distinct products, and where the components of the vector are a measure of the rating provided by that user for that product. Thus, for a set of m users, the user-product ratings matrix is an $m \times n$ matrix A, where A_{ij} is the rating by user i for product j (or is null if the rating is not provided). A recommendation algorithm generates recommendations for a new user based on a few users who are

most similar to the user after querying the new user about his (or her) rating on a small number of products. For more details, see [54, 10, 1].

A matrix CUR decomposition has been used to obtain competitive recommendation performance by judiciously sampling $O(m+n)$ entries of the user-product ratings matrix and reconstructing missing entries [20]. In more detail, assuming access to a matrix C consisting of the ratings of every user for a small number of products and a matrix R consisting of the ratings of a small number of users for every product, then, under assumptions, CUR is a provably good approximation to the user-product matrix A [20]. Prior theoretical work on recommendation systems includes Kumar et al. [42], who offer competitive algorithms even with only two samples/customer, assuming a strong clustering of the products; Azar et al. [4], who use spectral methods to recreate very accurately the user-product ratings matrix A, assuming a certain gap requirement and a sample of $\Omega(mn)$ entries of A; Kleinberg and Sandler [37], who develop recommendation algorithms with provable performance guarantees in a probabilistic mixture mode; and (most relevant for our work) Drineas, Kerenidis, and Raghavan [20], who obtain competitive performance by sampling $O(m + n)$ entries of the user-product ratings matrix and reconstructing missing entries with a matrix CUR decomposition. Other applications of linear algebra have used the SVD for dimensionality reduction [9, 58, 26].

Although the ratings in the user-product matrix A are often interpreted in terms of the utility of product j for user i, utility in neoclassical economics is an ordinal and not a cardinal concept. This is because utility functions are constructs that encode preference information and because the same preferences are described when the utility function is subject to a wide class of monotonic transformations. This observation motivates the definition of an $m \times n \times n$ user-product-product $(2 + 1)$-tensor $\mathcal{A}$, where $\mathcal{A}_{ijk}$ is $+1$ or -1 depending on whether product j or product k is preferred by user i. Similar preference-based models have appeared [12, 24, 35, 34] and have been motivated by such observations as that two users with very similar preferences for items may have very different rating schemes. When faced with a new user, this preference model depends on obtaining pairwise preference information such as that the user bought product `A` when he could have bought product `B` or that the user clicked on link `A` when he could have clicked on link `B`.

5.1. Description of data and the model. Under this preference model for recommendation system analysis, the tensor $\mathcal{C}$ consists of a small number of dictionary or basis elements from a small number of users, where each element corresponds to the full $n \times n$ pairwise preference matrix for a single user. Similarly, the matrix $\mathcal{R}$ consists of a dictionary or basis set of preference information from every user about a small number of product-product pairs.

In the next subsection, we will see that the tensor-CUR decomposition can be applied to recommendation system data under this model to reconstruct missing entries in the user-product-product preference tensor in order to make high-quality recommendations. Since most recommendation system databases do not provide data in this preference-based format, the data set we will consider will be derived from the ratings in the well-studied Jester data [26]. As an initial application, we consider the $m = 14{,}116$ (out of ca. 73,421) users who rated all of the $n = 100$ products (i.e., jokes). From this $m \times n$ user-product ratings matrix, we define an $m \times n \times n$ user-product-product preference tensor by performing the following for each user: convert the n-dimensional rating vector into an $n \times n$ preference matrix in which the ij entry is $+1$ or -1 depending on whether or not the user prefers product i to product j.

(Although this results in ordered and fully consistent preferences, this is not required by our decomposition.) In this application, in the absence of a better model, both slabs and fibers will be chosen uniformly at random.

5.2. Recommendation quality results. We now describe our results for the tensor-CUR decomposition when applied to the Jester dataset in the context of recommendation systems. Let c be an integer between 1 and 14,116 (recall that this is the total number of users that fully rated all 100 jokes in the Jester data), and assume that we sample uniformly at random c of the 14,116 users. For each sampled user, we assume that the corresponding 100×100 slab of the $100 \times 100 \times 14{,}116$ tensor representing the Jester dataset (see the previous section for details) is fully known or, in other words, that we know all pairwise product-product (i.e., joke-joke) comparisons for the c sampled users.

Consider the $14{,}116 - c$ slabs (i.e., users) that we did not sample. For each such *target* slab (i.e., *target* user), we use the tensor-CUR decomposition to reconstruct it as a linear combination of the c sampled slabs. Thus, it suffices to compute c coefficients such that a linear combination of the basis slabs using these coefficients achieves a satisfactory reconstruction of the target slab. However, in order to do such a reconstruction, we need some information from the target slab. This information consists of a small number of product-product preference queries sampled uniformly at random from the target slab. These elements of the target slab will allow us to approximately infer the coefficients to be used in expressing the target slab as a linear combination of the c basis slabs. Once the target slab (i.e., preference matrix) is reconstructed, we can use this reconstruction to make recommendations by picking the N products with the largest row sums. In our model, where the (i, j)th entry of the preference matrix is set to 1 if product i is preferred over product j and to -1 otherwise, such rows correspond to the most desirable products for this user.

To formally evaluate the quality of our recommender system, we use the well-known top-N procedure and compute the precision, recall, and the $F1$ statistic [58]. More specifically, let $\mathcal{T}_N$ be the actual set of the top N products for a certain user, and let $\mathcal{S}_K$ be a set of K products that are *suggested* to this user by a recommender system. Clearly, K can be equal to or larger than N, whereas values of K that are smaller than N are typically not interesting. For some combinations of N and K, we shall measure the following four quantities.

Successful recommendations. The number of elements in the intersection of $\mathcal{T}_N$ and $\mathcal{S}_K$ or, in other words, the number of products that are in the top-N preferred products for a particular user *and* were recommended by an algorithm that made K suggestions.

Recall. The number of successful recommendations divided by the number of suggestions (K) made by the algorithm. This quantity normalizes the number of successful recommendations to take into account the fact that increasing the number of suggestions increases the number of top-N products recommended by the algorithm.

Precision. The number of successful recommendations divided by N. Remember that N essentially determines the number of products that a user is interested in, and hence this quantity normalizes the number of successful recommendations to take into account the fact that increasing N increases the number of top-N products recommended by the algorithm.

F1 statistic. The formal definition is

$$\text{F1 statistic} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

and it is commonly used to reconcile the mutually conflicting nature of the precision and recall statistics. (Notice, for example, that increasing N tends to increase recall but decreases precision [58].)

Prior to presenting the results of our experimental evaluation, we briefly discuss our choices for the four parameters involved in our experiments. First, recall that c denotes the number of basis users that reveal all their pairwise product preferences to the algorithm; we let c be all powers of 2 between 2 and 1024. This choice provides a clear picture of the behavior of tensor-CUR for very small (e.g., $c \leq 32$), medium-sized (e.g., $64 \leq c \leq 256$), and large (e.g., $c = 512, 1024$) basis sets. Second, N is set to be either 5 or 10, implying the algorithm is successful if it recommends one of the top 5 or top 10 products for a certain user. Third, K is set to be equal to N or $2N$, and hence the algorithm is allowed to suggest either 5 or 10 products for the top-5 case and either 10 or 20 products for the top-10 case. Fourth, the number of fibers that the tensor-CUR algorithm samples or, in other words, the number of product-product pairwise comparisons of a target user that are revealed to the algorithm is again set to all powers of 2 between 2 and 1024; the rationale is the same as above. In the first experiment, we will set the number of fibers to $100^2 = 10{,}000$ (all available fibers), in order to illustrate the limiting behavior of tensor-CUR. We emphasize that both the sampling of slabs and the sampling of fibers are done uniformly at random without replacement, and hence sampling 10,000 fibers is equivalent to picking all the fibers.

In our first experiment, we seek to determine an upper bound on the quality of recommendations based on using a small number of basis slabs and all fibers for the remaining users. Clearly, this experiment seeks only to characterize the limiting behavior of tensor-CUR, since having all fibers trivially allows perfect recommendations. Figures 11 and 12 illustrate that almost all users can be very accurately expressed as a linear combination of a small number of basis users, chosen uniformly at random without replacement. In later experiments, given this observation, we will attempt to approximate the coefficients of this linear combination using a small number of fibers.

Figure 11 shows the results for $N = 5$. Notice that using 512 or 1024 slabs and only 5 suggestions results in 4 or more successful recommendations; if the algorithm is allowed to make 10 suggestions, 64 or more slabs are enough to make roughly four successful recommendations. (As a trivial but weak lower bound on quality, by making five suggestions uniformly at random, we expect that we will make ca. .5 predictions correctly, since we are making 5 predictions and there are 100 products.) Notice that the $F1$ statistic shows a change of phase as the number of slabs increases above 256: making more than 5 suggestions is not necessary anymore, since the number of basis slabs suffices to accurately capture the high-ranking products. Given less than 256 basis slabs (e.g., 128 slabs), our results suggest that making 10 suggestions is qualitatively better. The same conclusions essentially apply to Figure 12 as well, which shows the results for $N = 10$. However, we should emphasize that the effect of making 20 versus 10 suggestions, as measured by the $F1$ statistic, is much less obvious in this case. Notice that making 20 suggestions does not result in a significant advantage even for a small number of basis slabs and is clearly worse as the number of basis slabs increases above 128.

In our second experiment, we show that by using a basis of preference information from (say) 128 users and performing a small number of product-product preference queries on a new user, we can make a large number of high-quality recommendations both for the top-5 and top-10 cases; see Figures 13 and 14, respectively. Since we are sampling a small number of fibers in this case, we are performing an approximate least-squares fit using just the information about a new user contained in a small number

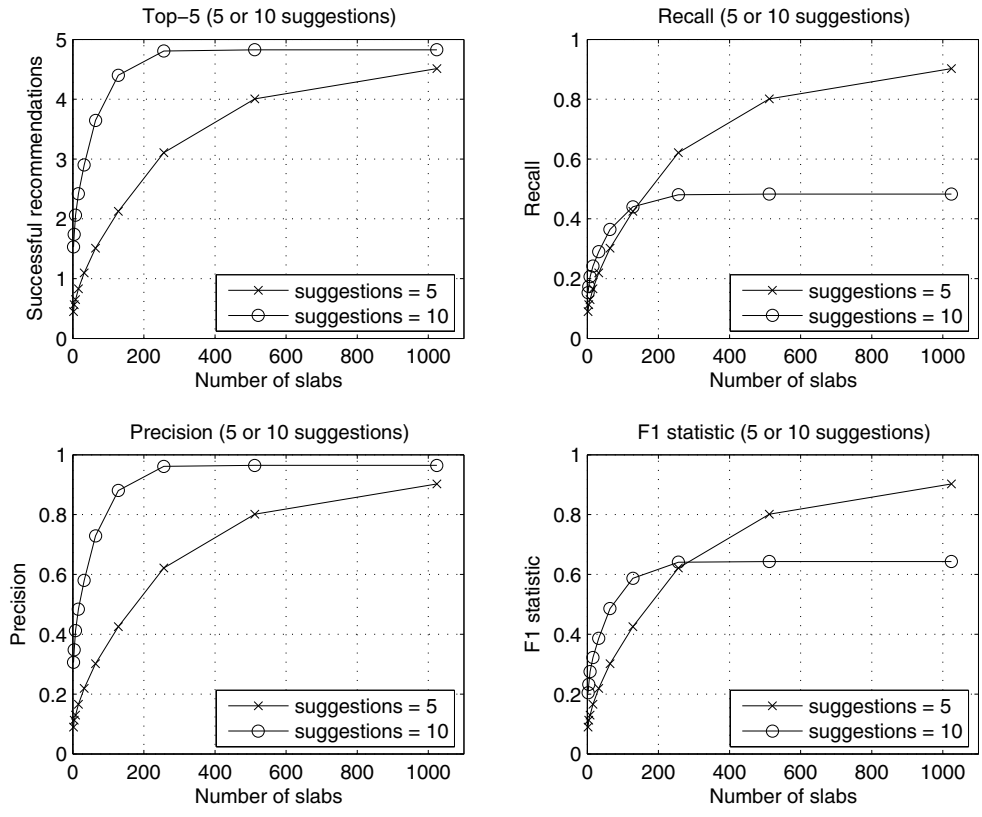


FIG. 11. *Effect of user basis size on top-5 recommendation quality using complete pairwise product-product preference information. The basis users are sampled uniformly at random without replacement.*

of fibers. If the algorithm is allowed to make 10 suggestions, the statistics for top-5 recommendations remain competitive with the upper bounds suggested in Figure 11. However, if the algorithm is allowed only 5 suggestions, the results are markedly worse, especially given a small number of pairwise product-product comparisons. Naturally, the $F1$ statistic illustrates that suggesting 10 products is now always preferable to suggesting 5 products. This observation changes when we evaluate the algorithm on top-10 recommendations, where the $F1$ statistic shows that suggesting 10 or 20 products is essentially the same, and thus suggesting 10 products is the right course of action. Notice that even though the performance of the algorithm is worse than the optimal one of Figure 12, it is clearly well above the random level. We would also like to note the nonmonotonicity near ca. 64 queries; this seems to be a fitting issue. Figures 15 and 16 show the results for top-10 recommendations when the number of basis users is set to 64 and 256, respectively. The results are qualitatively similar, but it is worth noticing that the algorithm making 10 suggestions outperforms the algorithm making 20 suggestions given 256 basis slabs and more than 256 fibers. The results for top-5 recommendations using 64 and 256 basis users are omitted, since they are qualitatively the same as in Figure 11.

In our third and final experiment, we present the distribution of correct top-10 predictions for the 14,116 users by using 64 or 128 basis users and a variable number

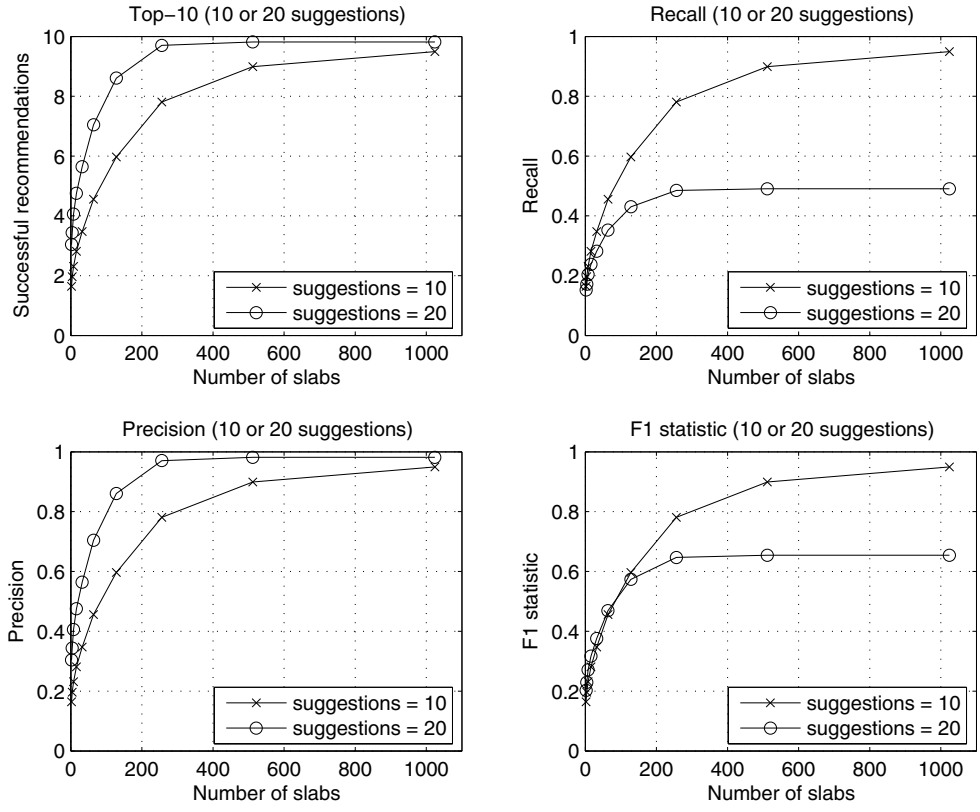


FIG. 12. *Effect of user basis size on top-*10 *recommendation quality using complete pairwise product-product preference information. The basis users are sampled uniformly at random without replacement.*

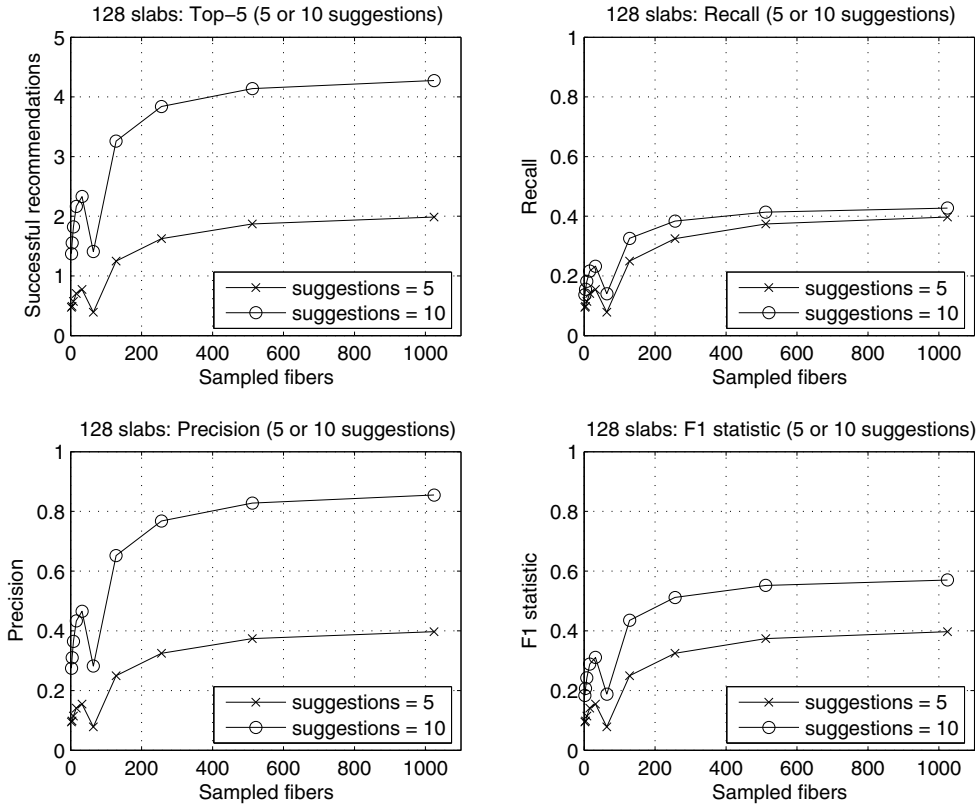


FIG. 13. *Effect of number of sampled fibers (pairwise product-product comparisons) on the top-5 recommendation quality given 128 basis users, sampled uniformly at random without replacement. The fibers are also sampled uniformly at random without replacement.*

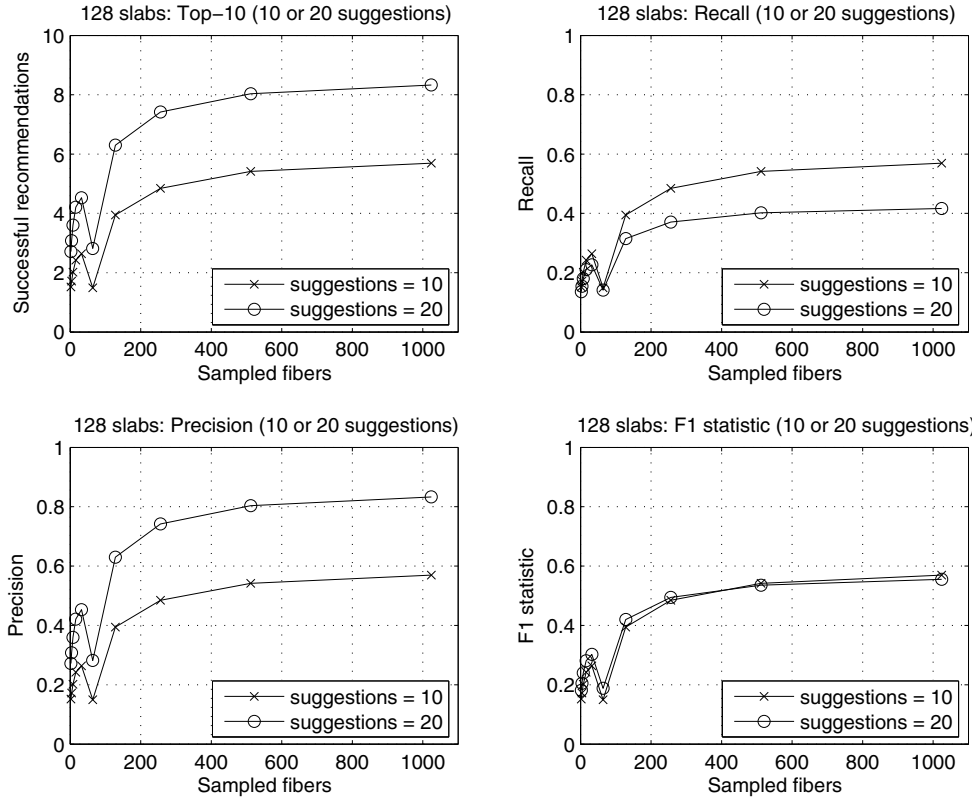


FIG. 14. *Effect of number of sampled fibers (pairwise product-product comparisons) on the top-*10 *recommendation quality given* 128 *basis users, sampled uniformly at random without replacement. The fibers are also sampled uniformly at random without replacement.*

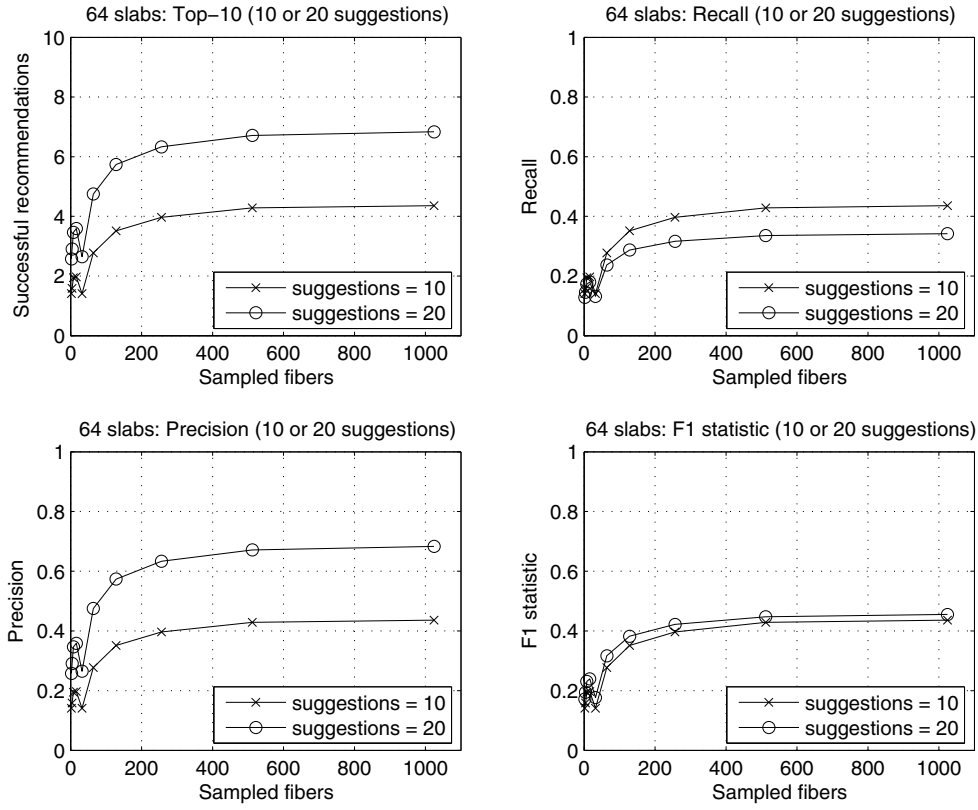


FIG. 15. *Effect of number of sampled fibers (pairwise product-product comparisons) on the top-10 recommendation quality given 64 basis users, sampled uniformly at random without replacement. The fibers are also sampled uniformly at random without replacement.*

of pairwise product-product comparisons; see Figure 17. Clearly, as the number of basis slabs or sampled fibers increases, the curves are shifted to the right, illustrating that a larger number of users receives more accurate recommendations. In this case, we plot results for the algorithm making 10 suggestions. Similar results are seen in all other cases.

In evaluating performance, we distinguish between prediction and reconstruction. In the former, we want to know how much user i will like product j (in a ratings model) or whether user i will prefer product j or product k (in a preference model). In the latter, which is of interest to us, we want to give a list of, e.g., the top-10 products for user i. We use tensor reconstruction as an intermediate step to making high-quality recommendations.

6. Conclusion. We have developed a tensor-based extension of the matrix CUR decomposition. This tensor-CUR decomposition is of most interest when the data may be modeled by a variable subscripted by three or more indices and when one of those indices/modes is qualitatively different from the others. In this case, the tensor-CUR decomposition approximately expresses the original data tensor in terms of a basis consisting of underlying subtensors that are actual data elements and thus that have natural interpretation in terms of the processes generating the data. In addition, we have applied the tensor-CUR decomposition to problems in two diverse domains of

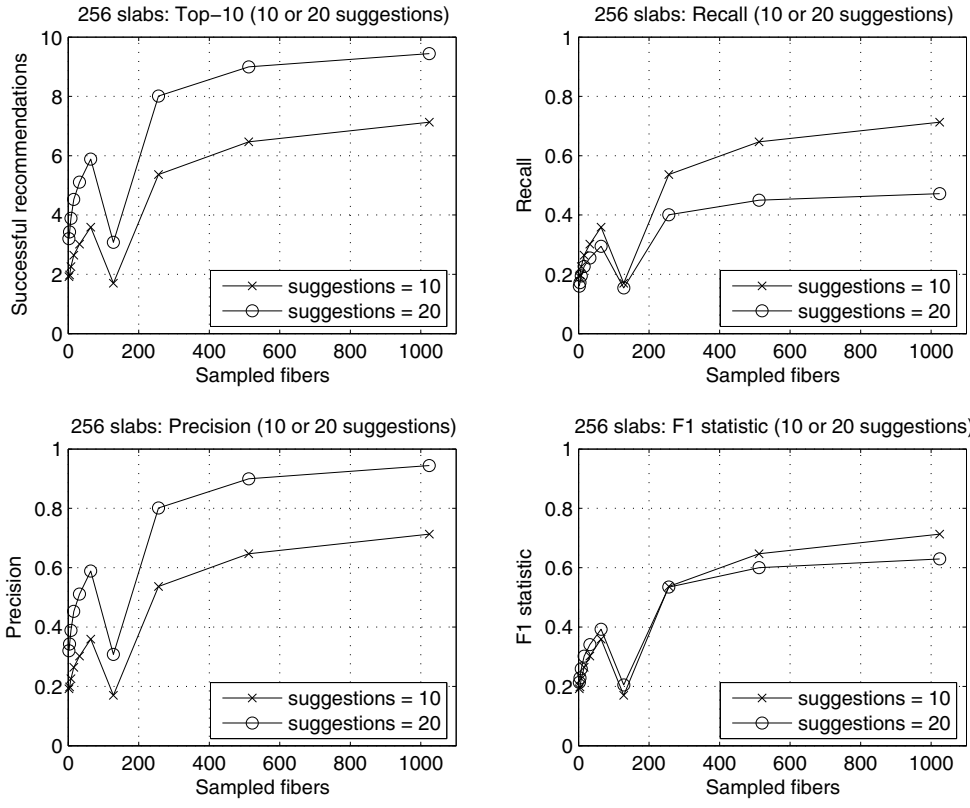


FIG. 16. *Effect of number of sampled fibers (pairwise product-product comparisons) on the top-10 recommendation quality given 256 basis users, sampled uniformly at random without replacement. The fibers are also sampled uniformly at random without replacement.*

data analysis: hyperspectral medical image analysis and consumer recommendation system analysis.

Similarities and differences between the methods discussed in this paper and the image analysis techniques known as "eigenfaces" and "tensor-faces" should be mentioned. The method of eigenfaces computes the eigenvectors of the covariance matrix of a large number of images of faces [62]. Eigenanalysis (and, more generally, SVD analysis) successively computes axes of maximum variation in the data, conditioned on being orthogonal to previously computed axes. Since this orthogonality is not present in natural images of faces, its imposition results in the characteristic "ringing" oscillations generated by eigenanalysis of facial images that in turn leads to difficulty interpreting the eigenfaces after the first few. The methods of the present paper are applicable to a set of time-resolved or frequency-resolved images of a single object. One could apply SVD-type analysis for data compression, i.e., to reduce the dimensionality along the slabs and/or the fibers. On the other hand, it will likely be difficult to interpret the principal components. Our tensor-CUR algorithms provide approximate low-rank tensor decompositions in terms of actual data elements. If orthogonality is not present in the data, e.g., if there are different fibers and/or pixels, then the tensor-CUR decompositions will be in terms of nonorthogonal data elements. Partly in response to ringing artifacts of eigenface analysis, a tensor-based

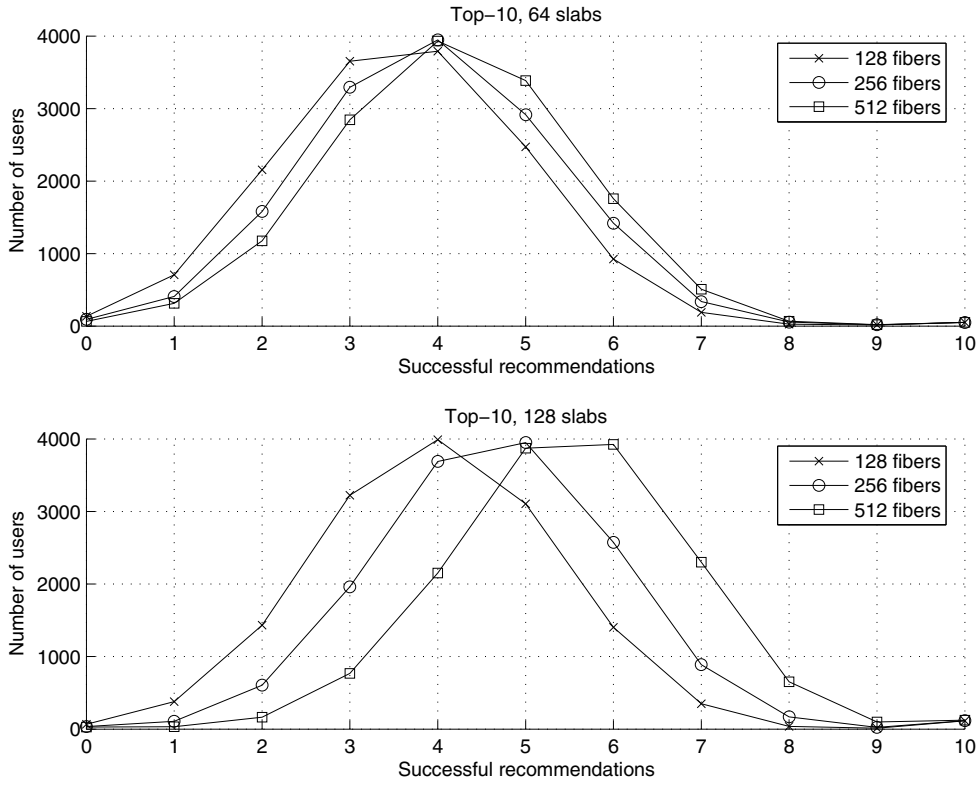


FIG. 17. *Distribution of number of users getting a given number of successful top-*10 *recommendations for a basis consisting of* 64 *or* 128 *users for different numbers of sampled fibers. Both the basis slabs and the fibers are sampled uniformly at random without replacement.*

analysis of facial images has been introduced [65]. This analysis involves applying a tensor-based generalization of the SVD to a user-defined set of features derived from a set of images of faces. A randomized variant of this generalization has been presented and analyzed in [23]. This randomized tensor-SVD algorithm bears some similarity to the randomized tensor-CUR algorithms described in this paper. It differs, however, in that there is no preferred mode; instead, the tensor is "unfolded" along every mode, and a projection along each mode is constructed by sampling columns along that mode.

We conclude with several related extensions of the present work. First, it would be worth examining how these methods can be coupled with more traditional methods of image analysis and recommendation system analysis. This could be performed either by choosing slabs and fibers and then analyzing each slab or fiber with more traditional methods, or by using structural insights from more traditional methods to construct the sample of slabs and fibers, or by compressing each individual slab with more traditional methods. Second, it would be worth determining whether the sample of slabs and/or fibers could be chosen to preserve some interesting multilinear structure in the data tensors that is damaged by the sampling techniques we have used. Third, it would be worth determining the extent to which it would be possible to combine fibers from several data cubes into a "dictionary" that could be used, along with a few slabs in a new data cube, to describe the entire new data cube. Fourth, it

would be worth understanding in greater detail the relationship between the methods we have presented for analyzing tensor data and the well-studied model proposed by Tucker, the "canonical decomposition" model, the "parallel factors" model, and the higher-order SVD model; due to lack of space, a comparison with these models has been omitted. Finally, cross-approximation techniques are powerful and well-developed adaptive methods for low-rank approximation of matrices [6, 63]; it is worth understanding in greater detail the relationship between these methods and matrix CUR decompositions.

Acknowledgment. We thank the authors of [51], in particular Gustave L. Davis of Yale University, for making available the hyperspectral data.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*, IEEE Trans. Knowl. Data Eng., 17 (2005), pp. 734–749.

[2] C. Angeletti, N. R. Harvey, V. Khomitch, R. Levenson, and D. L. Rimm, *Detection of malignant cells in cytology specimen using genie hybrid genetic algorithm*, Mod. Pathol., 17 (2004), Suppl 1:350A.

[3] C. Angeletti, R. Jaganth, R. M. Levenson, and D. L. Rimm, *Spectral analysis: A novel method for classification of urine cytology.* Mod. Pathol., 16 (2003), 57A.

[4] Y. Azar, A. Fiat, A. R. Karlin, F. McSherry, and J. Saiaz, *Spectral analysis of data*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, 2001, pp. 619–626.

[5] T. S. Barry, A. M. Gown, H. E. Yaziji, and R. W. Levenson, *Use of spectral imaging analysis for evaluation of multi-color immuno-histochemistry*, Mod. Pathol., 17 (2004), Suppl 1:350A.

[6] M. Bebendorf, *Approximation of boundary element matrices*, Numer. Math., 86 (2000), pp. 565–589.

[7] M. W. Berry, S. A. Pulatova, and G. W. Stewart, *Computing Sparse Reduced-Rank Approximations to Sparse Matrices*, Technical report UMIACS TR-2004-32 CMSC TR-4589, University of Maryland, College Park, MD, 2004.

[8] G. Beylkin and M. J. Mohlenkamp, *Numerical operator calculus in higher dimensions*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 10246–10251.

[9] D. Billsus and M. J. Pazzani, *Learning collaborative information filters*, in Proceedings of the 15th International Conference on Machine Learning, Morgan Kaufman, San Francisco, 1998, pp. 46–54.

[10] J. Breese, D. Heckerman, and C. Kadie, *Empirical analysis of predictive algorithms for collaborative filtering*, in Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence, Morgan Kaufman, San Francisco, 1998, pp. 43–52.

[11] J. D. Carroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[12] W. W. Cohen, R. E. Schapire, and Y. Singer, *Learning to order things*, in Annual Advances in Neural Information Processing Systems 10: Proceedings of the 1997 Conference, 1998, pp. 451–457.

[13] R. R. Coifman, *Multiresolution analysis in non-homogeneous media*, in Wavelets. Time-Frequency Methods and Phase Space, J.-M. Combes, A. Grossmann, and P. Tchamitchian, eds., Springer-Verlag, Berlin, 1989, p. 259.

[14] G. L. Davis, M. Maggioni, R. R. Coifman, D. L. Rimm, and R. M. Levenson, *Spectral/spatial analysis of colon carcinoma*, Mod. Pathol., 16 (2003), 3320:3321A.

[15] G. L. Davis, M. Maggioni, F. J. Warner, F. B. Geshwind, A. C. Coppi, R. A. DeVerse, and R. R. Coifman, *Spectral analysis of normal and malignant microarray tissue sections using a novel micro-optoelectricalmechanical system*, Mod. Pathol., 17 (2004), 1:358A.

[16] R. A. DeVerse, R. R. Coifman, A. C. Coppi, W. G. Fateley, F. Geshwind, R. M. Hammaker, S. Valenti, F. J. Warner, and G. L. Davis, *Application of spatial light modulators for new modalities in spectrometry and imaging*, in Proceedings of the SPIE 4959, 2003, pp. 12–22.

[17] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices* I: *Approximating matrix multiplication*, SIAM J. Comput., 36 (2006), pp. 132–157.
[18] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices* II: *Computing a low-rank approximation to a matrix*, SIAM J. Comput., 36 (2006), pp. 158–183.
[19] P. Drineas, R. Kannan, and M. W. Mahoney, *Fast Monte Carlo algorithms for matrices* III: *Computing a compressed approximate matrix decomposition*, SIAM J. Comput., 36 (2006), pp. 184–206.
[20] P. Drineas, I. Kerenidis, and P. Raghavan, *Competitive recommendation systems*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, 2002, pp. 82–90.
[21] P. Drineas and M. W. Mahoney, *Approximating a Gram matrix for improved kernel-based learning*, in Proceedings of the 18th Annual Conference on Learning Theory, Springer-Verlag, Berlin, 2005, pp. 323–337.
[22] P. Drineas and M. W. Mahoney, *On the Nyström method for approximating a Gram matrix for improved kernel-based learning*, J. Mach. Learn. Res., 6 (2005), pp. 2153–2175.
[23] P. Drineas and M. W. Mahoney, *A randomized algorithm for a tensor-based generalization of the singular value decomposition*, Linear Algebra Appl., 420 (2007), pp. 553–571.
[24] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, *An efficient boosting algorithm for combining preferences*, J. Mach. Learn. Res., 4 (2003), pp. 933–969.
[25] Y. Garini, N. Katzir, D. Cabib, R. A. Buckwald, D.G. Soenksen, and Z. Malik, *Spectral bio-imaging*, in Fluorescence Imaging Spectroscopy and Microscopy, John Wiley and Sons, New York, 1996, pp. 87–124.
[26] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, *Eigentaste: A constant time collaborative filtering algorithm*, Inform. Retrieval, 4 (2001), pp. 133–151.
[27] T. Gonzalez and J. Ja'Ja', *On the complexity of computing bilinear forms with* $\{0, 1\}$ *constants*, J. Comput. System Sci., 20 (1980), pp. 77–95.
[28] S. A. Goreinov and E. E. Tyrtyshnikov, *The maximum-volume concept in approximation by low-rank matrices*, in Structured Matrices in Mathematics, Computer Science, and Engineering, I., Contemp. Math. 280, AMS, Providence, RI, 2001, pp. 47–51.
[29] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21.
[30] W. H. Greub, *Multilinear Algebra*, Springer-Verlag, Berlin, 1967.
[31] R. A. Harshman and M. E. Lundy, *The PARAFAC model for three-way factor analysis and multidimensional scaling*, in Research Methods for Multimode Data Analysis, H. G. Law, C. W. Snyder, Jr., J. Hattie, and R. P. McDonald, eds., Praeger, New York, 1984, pp. 122–215.
[32] J. Håstad, *Tensor rank is NP-complete*, J. Algorithms, 11 (1990), pp. 644–654.
[33] T. D. Howell, *Global properties of tensor rank*, Linear Algebra Appl., 22 (1978), pp. 9–23.
[34] R. Jin, L. Si, and C. X. Zhai, *Preference-based graphic models for collaborative filtering*, in Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence, Morgan Kaufman, San Francisco, 2003, pp. 329–336.
[35] R. Jin, L. Si, C. X. Zhai, and J. Callan, *Collaborative filtering with decoupled models for preferences and ratings*, in Proceedings of the 12th ACM International Conference on Information and Knowledge Management, 2003, pp. 309–316.
[36] H. A. L. Kiers, *Towards a standardized notation and terminology in multiway analysis*, J. Chemometrics, 14 (2000), pp. 105–122.
[37] J. Kleinberg and M. Sandler, *Using mixture models for collaborative filtering*, in Proceedings of the 36th Annual ACM Symposium on Theory of Computing, 2004, pp. 569–578.
[38] T. G. Kolda, *A counterexample to the possibility of an extension of the Eckart–Young low-rank approximation theorem for the orthogonal rank tensor decomposition*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 762–767.
[39] P. M. Kroonenberg and J. De Leeuw, *Principal component analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 45 (1980), pp. 69–97.
[40] J. B. Kruskal, *Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.
[41] J. B. Kruskal, *Rank, decomposition, and uniqueness for* 3*-way and* N*-way arrays*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North–Holland, Amsterdam, 1989, pp. 7–18.
[42] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, *Recommendation systems: A probabilistic analysis*, in Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, 1998, pp. 664–673.

[43] L. De Lathauwer, B. De Moor, and J. Vandewalle, *An introduction to independent component analysis*, J. Chemometrics, 14 (2000), pp. 123–149.

[44] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[45] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[46] D. Leibovici and R. Sabatier, *A singular value decomposition of a k-way array for a principal component analysis of multiway data, PTA-k*, Linear Algebra Appl., 269 (1998), pp. 307–329.

[47] R. M. Levenson and D. Farkas, *Digital spectral imaging for histopathology and cytopathology*, in Proceedings of the SPIE 2983, 1997, pp. 123–135.

[48] L.-H. Lim and G. H. Golub, *Tensors for Numerical Multilinear Algebra: Ranks and Basic Decompositions*, Technical report 05-02, Stanford University SCCM, Stanford, CA, 2005.

[49] G. Linden, B. Smith, and J. York, *Amazon.com recommendations: Item-to-item collaborative filtering*, IEEE Internet Comput., 7 (2003), pp. 76–80.

[50] C. F. Van Loan, *The ubiquitous Kronecker product*, J. Comput. Appl. Math., 123 (2000), pp. 85–100.

[51] M. Maggioni, G. L. Davis, F. J. Warner, F. B. Geshwind, A. C. Coppi, R. A. DeVerse, and R. R. Coifman, *Algorithms from Signal and Data Processing Applied to Hyperspectral Analysis: Discriminating Normal and Malignant Microarray Colon Tissue Sections Using a Novel Digital Mirror Device System*, Technical report YALEU/DCS/TR-1311, Yale University Department of Computer Science, New Haven, CT, 2004.

[52] M. W. Mahoney, M. Maggioni, and P. Drineas, *Tensor-CUR decompositions for tensor-based data*, in Proceedings of the 12th Annual ACM SIGKDD Conference, 2006, pp. 327–336.

[53] A. Papadakis, E. Stathopoulos, G. Delides, K. Berberides, G. Nikiforides, and C. Balas, *A novel spectral microscope system: Application in quantitative pathology*, IEEE Trans. Biomed. Eng., 50 (2003), pp. 207–217.

[54] P. Resnick and H. R. Varian, *Recommender systems*, Comm. ACM, 40 (1997), pp. 56–58.

[55] C. Rothman, I. Bar-Am, and Z. Malik, *Spectral imaging for quantitative histology and cytogenetics*, Histology and Histopathology, 13 (1998), pp. 921–926.

[56] N. Saito and R. R. Coifman, *Local discriminant bases and their applications*, J. Math. Imaging Vision, 5 (1995), pp. 337–358.

[57] N. Saito, R. R. Coifman, F. B. Geshwind, and F. Warner, *Discriminant feature extraction using empirical probability density estimation and a local basis library*, Pattern Recognition, 35 (2002), pp. 2841–2852, 2002.

[58] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, *Application of dimensionality reduction in recommender system—a case study*, in Proceedings of the WebKDD 2000 Workshop, 2000.

[59] G. W. Stewart, *Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix*, Numer. Math., 83 (1999), pp. 313–323.

[60] G. W. Stewart, *Error analysis of the quasi-Gram–Schmidt algorithm*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 493–506.

[61] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[62] M. Turk and A. Pentland, *Eigenfaces for recognition*, J. Cogn. Neurosci., 3 (1991), pp. 71–96.

[63] E. Tyrtyshnikov, *Incomplete cross approximation in the mosaic-skeleton method*, Computing, 64 (2000), pp. 367–380.

[64] S. G. Vari, G. Muller, J. M. Lerner, and R. D. Naber, *Telepathology and imaging spectroscopy as a new modality in histopathology*, Stud. Health Technol. Inform., 68 (1999), pp. 211–216.

[65] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear analysis of image ensembles: TensorFaces*, in Proceedings of the 7th European Conference on Computer Vision, Springer-Verlag, Berlin, 2002, pp. 447–460.

[66] C. K. I. Williams and M. Seeger, *Using the Nyström method to speed up kernel machines*, in Annual Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference, MIT Press, Cambridge, MA, 2001, pp. 682–688.

[67] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 988–1007

LOW-RANK APPROXIMATION OF GENERIC $p \times q \times 2$ ARRAYS AND DIVERGING COMPONENTS IN THE CANDECOMP/PARAFAC MODEL*

ALWIN STEGEMAN†

Abstract. We consider the low-rank approximation over the real field of generic $p \times q \times 2$ arrays. For all possible combinations of p, q, and R, we present conjectures on the existence of a best rank-R approximation. Our conjectures are motivated by a detailed analysis of the boundary of the set of arrays with at most rank R. We link these results to the Candecomp/Parafac (CP) model for three-way component analysis. Essentially, CP tries to find a best rank-R approximation to a given three-way array. In the case of $p \times q \times 2$ arrays, we show (under some regularity condition) that if a best rank-R approximation does not exist, then any sequence of CP updates will exhibit diverging CP components, which implies that several components are highly correlated in all three modes and their component weights become arbitrarily large. This extends Stegeman [*Psychometrika*, 71 (2006), pp. 483–501], who considers $p \times p \times 2$ arrays of rank $p+1$ or higher. We illustrate our results by means of simulations.

Key words. low-rank tensor approximations, border rank, arrays, Candecomp, Parafac, three-way arrays, degenerate Parafac solutions

AMS subject classifications. 15A03, 15A22, 15A69, 49M27, 62H25

DOI. 10.1137/050644677

1. Introduction.

We consider the problem of finding a best low-rank approximation to a three-way array $\underline{\mathbf{X}} \in \mathbb{R}^{p \times q \times 2}$. In this introductory section, we discuss the general problem of finding a best low-rank approximation to a k-way array and its applications in algebraic complexity theory (the multiplicative complexity of the computation of bilinear forms) and psychometrics (the Candecomp/Parafac (CP) model for three-way component analysis). Also, the consequences of an array not having a best low-rank approximation are discussed for these fields of research. Finally, we consider some results of the theory of matrix pencils with implications on the rank of $p \times q \times 2$ arrays, and show how our analysis fits into this literature.

1.1. Low-rank approximation of arrays.

Let the rank over a field $\mathcal{F}$ of a k-way array $\underline{\mathbf{X}} \in \mathcal{F}^{d_1 \times \cdots \times d_k}$ be defined in the usual way, i.e., as the smallest number of rank-1 arrays in $\mathcal{F}^{d_1 \times \cdots \times d_k}$ whose sum equals $\underline{\mathbf{X}}$; see Hitchcock [15, 16]. A k-way array has rank 1 over $\mathcal{F}$ if it is the outer product of k vectors in $\mathcal{F}^{d_1}, \ldots, \mathcal{F}^{d_k}$. The problem of finding a best rank-R approximation of $\underline{\mathbf{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ boils down to minimizing

$$\left\| \underline{\mathbf{X}} - \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(k)} \right\| \tag{1.1}$$

over the vectors $\mathbf{a}_r^{(j)} \in \mathcal{F}^{d_j}$, $j = 1, \ldots, k$, $r = 1, \ldots, R$, where $\circ$ denotes the outer product and $\|\cdot\|$ denotes some norm on $\mathcal{F}^{d_1 \times \cdots \times d_k}$. Unless stated otherwise, we will

*Received by the editors November 9, 2005; accepted for publication (in revised form) by L. De Lathauwer March 6, 2007; published electronically September 25, 2008. This research was supported by the Dutch Organisation for Scientific Research (NWO), VENI grant 451-04-102.
http://www.siam.org/journals/simax/30-3/64467.html
†Heijmans Institute of Psychological Research, University of Groningen, Grote Kruisstraat 2/1, 9712 TS Groningen, The Netherlands (a.w.stegeman@rug.nl).

assume that $\mathcal{F} = \mathbb{R}$ and $\|\cdot\|$ is the Frobenius norm. We denote the rank of an array $\underline{\mathbf{X}}$ as $\text{rank}_\circ(\underline{\mathbf{X}})$.

For $k = 2$, all best rank-R approximations can be obtained from the singular value decomposition of the matrix to be approximated; see Eckart and Young [11]. However, for $k \geq 3$ a best rank-R approximation does not always exist. Examples of arrays that can be approximated arbitrarily well by arrays of lower rank are known from the algebraic complexity literature; see Bini et al. [1], Bini, Lotti, and Romani [2], and Bini [3], as well as from the psychometric and chemometric literature; see ten Berge, Kiers, and De Leeuw [37] and Paatero [30]. Stegeman [34] has shown, under some regularity condition, that generic $p \times p \times 2$ arrays of rank $p+1$ (a set of positive volume in $\mathbb{R}^{p\times p\times 2}$) do not have a best rank-p approximation. De Silva and Lim [10] show that a best rank-1 approximation always exists, while for any $k \geq 3$, any $d_1, \ldots, d_k \geq 2$, and any $R \in \{2, \ldots, \min(d_1, \ldots, d_k)\}$, a rank-$(R+1)$ array exists which has no best rank-R approximation. Also, [10] show that all $2 \times 2 \times 2$ arrays of rank 3 (a set of positive volume in $\mathbb{R}^{2\times 2\times 2}$) have no best rank-2 approximation, and that, for any $d_1, d_2, d_3 \geq 2$, the set of arrays in $\mathbb{R}^{d_1\times d_2\times d_3}$ which have no best rank-2 approximation has positive volume.

1.2. Algebraic complexity theory and array rank. An important problem in algebraic complexity theory is the multiplicative complexity of the computation of a set of bilinear forms $\mathbf{u}^T\mathbf{X}_k\mathbf{v}$, $k = 1, \ldots K$, where $\mathbf{u}$ and $\mathbf{v}$ are indeterminates and the $\mathbf{X}_k$ have elements in a field $\mathcal{F}$. Strassen [36] showed that the K bilinear forms $\mathbf{u}^T\mathbf{X}_k\mathbf{v}$ can be computed with R nonscalar multiplications (i.e., multiplications of two elements not in $\mathcal{F}$), where R is the rank over $\mathcal{F}$ of the array $\underline{\mathbf{X}}$ with slices $\mathbf{X}_k$. Indeed, if $\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$, for vectors $\mathbf{a}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r$ with elements in $\mathcal{F}$, then we have

$$\mathbf{u}^T\mathbf{X}_k\mathbf{v} = \sum_{r=1}^{R} (\mathbf{u}^T\mathbf{a}_r)\,(\mathbf{b}_r^T\mathbf{v})\,c_{kr}\,, \tag{1.2}$$

and the K bilinear forms $\mathbf{u}^T\mathbf{X}_k\mathbf{v}$ can be computed using R nonscalar multiplications. See also Brockett and Dobkin [6].

Suppose $\mathcal{F} = \mathbb{R}$ and $\underline{\mathbf{X}}$ can be approximated arbitrarily well by rank-$(R-1)$ arrays. In that case, we could replace $\underline{\mathbf{X}}$ in (1.2) by a rank-$(R-1)$ array close to it and use only $R-1$ nonscalar multiplications in the computation of the bilinear form. Since the rank-$(R-1)$ array can be chosen arbitrarily close to $\underline{\mathbf{X}}$, the error in the computation of the bilinear form can be made arbitrarily small. This idea was pointed out by Bini et al. [1], Bini, Lotti, and Romani [2], and Bini [3, 4] and has been used in the design of algorithms for matrix multiplication; see Bürgisser, Clausen, and Shokrollahi [7, Chapter 15] and the references therein. Hence, the nonexistence of a best rank-$(R-1)$ approximation of $\underline{\mathbf{X}}$ yields a faster and arbitrarily accurate computation of the bilinear forms.

To express the optimal computational gain that can be achieved by approximating $\underline{\mathbf{X}}$ with arrays of lower rank, Bini, Lotti, and Romani [2] have introduced the notion of *border rank*. The border rank of an array $\underline{\mathbf{X}}$, which we denote by $\text{rank}_B(\underline{\mathbf{X}})$, is defined as

$$\text{rank}_B(\underline{\mathbf{X}}) = \min\{R : \ \underline{\mathbf{X}} \text{ can be approximated arbitrarily well by arrays of rank } R\}\,. \tag{1.3}$$

From this definition it follows that $\text{rank}_B(\underline{\mathbf{X}}) \leq \text{rank}_\circ(\underline{\mathbf{X}})$. Results on the border rank of various arrays have been obtained by Bini [4, 5] and Landsberg [25]. For later

use we state the following result. Let $\text{rank}_i(\underline{\mathbf{X}})$ denote the rank of the set of mode i fibers of $\underline{\mathbf{X}}$, where a mode i fiber is a vector obtained by varying the mode i index and keeping all other indices fixed. This notion of rank is due to Hitchcock [15, 16] and the set of $\text{rank}_i(\underline{\mathbf{X}})$ for all i is called the *multilinear rank* of $\underline{\mathbf{X}}$ in De Silva and Lim [10]

PROPOSITION 1.1. *For a k-way array $\underline{\mathbf{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_k}$, there holds*

$$\max_i \text{rank}_i(\underline{\mathbf{X}}) \leq \text{rank}_B(\underline{\mathbf{X}}) \leq \text{rank}_\circ(\underline{\mathbf{X}}) \,. \tag{1.4}$$

Proof. The second inequality follows from (1.3). A proof of the inequality $\text{rank}_i(\underline{\mathbf{X}}) \leq \text{rank}_\circ(\underline{\mathbf{X}})$ is given by De Silva and Lim [10]. We state it here for completeness. Let $R = \text{rank}_\circ(\underline{\mathbf{X}})$ and $\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(k)}$. Then all mode i fibers lie in the span of $\mathbf{a}_1^{(i)}, \ldots, \mathbf{a}_R^{(i)}$. This implies $\text{rank}_i(\underline{\mathbf{X}}) \leq R$.

We show the first inequality by contradiction. Suppose $t = \text{rank}_B(\underline{\mathbf{X}}) < \text{rank}_i(\underline{\mathbf{X}})$. Then there exists a sequence of rank-t arrays $\underline{\mathbf{Y}}^{(n)}$ converging to $\underline{\mathbf{X}}$. But then also the matrices $\mathbf{Y}_i^{(n)}$, containing as columns the mode i fibers of $\underline{\mathbf{Y}}^{(n)}$, must converge to the matrix $\mathbf{X}_i$ containing the mode i fibers of $\underline{\mathbf{X}}$. This is a contradiction, since $\text{rank}(\mathbf{Y}_i^{(n)}) \leq t < \text{rank}(\mathbf{X}_i)$ for all n, and a matrix cannot be approximated arbitrarily well by matrices of lower rank. Note that the upper semicontinuity of the multilinear rank is used here; see De Silva and Lim [10]. ∎

1.3. The CP model and diverging components. Carroll and Chang [9] and Harshman [13] have independently proposed the same method for component analysis of three-way data arrays and named it Candecomp and Parafac, respectively. We denote the CP model as

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} \omega_r \, (\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r) + \underline{\mathbf{E}} \,, \tag{1.5}$$

where $\underline{\mathbf{X}}$ is a $d_1 \times d_2 \times d_3$ data array, ω_r is the weight of component r, and $\|\mathbf{a}_r\| = \|\mathbf{b}_r\| = \|\mathbf{c}_r\| = 1$ for $r = 1, \ldots, R$. The Frobenius norm of $\underline{\mathbf{E}}$ is minimized to find the R components $\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ and the weights ω_r. For an overview and comparison of CP algorithms, see Hopke et al. [17] and Tomasi and Bro [41]. From (1.5) it is clear that the CP model tries to find a best rank-R approximation to the three-way array $\underline{\mathbf{X}}$.

The CP model (1.5) can be seen as a three-way extension of the principal component analysis model for matrices. For example, if the vectors $\mathbf{a}_r$ are interpreted as the components in mode 1, then $\mathbf{b}_r$ and $\mathbf{c}_r$ are the loadings on these components for modes 2 and 3, respectively. The real-valued CP model, i.e., where $\underline{\mathbf{X}}$ and the model parameters are real-valued, is used in a majority of applications in psychometrics and chemometrics; see Kroonenberg [21] and Smilde, Bro, and Geladi [33]. Complex-valued applications of CP occur in, e.g., signal processing and telecommunications research; see Sidiropoulos [32]. In this paper, we consider only the real-valued CP model.

A matrix notation of the CP model (1.5) is as follows. Let $\mathbf{X}_k$ $(d_1 \times d_2)$ and $\mathbf{E}_k$ $(d_1 \times d_2)$ denote the kth slices of $\underline{\mathbf{X}}$ and $\underline{\mathbf{E}}$, respectively. Then (1.5) can be written as

$$\mathbf{X}_k = \mathbf{A} \, \mathbf{C}_k \, \mathbf{\Omega} \, \mathbf{B}^T + \mathbf{E}_k \,, \qquad k = 1, \ldots K \,, \tag{1.6}$$

where $\mathbf{A}$ $(d_1 \times R)$ and $\mathbf{B}$ $(d_2 \times R)$ have the vectors $\mathbf{a}_r$ and $\mathbf{b}_r$ as columns, respectively, $\mathbf{\Omega}$ $(R \times R)$ is the diagonal matrix with the weights ω_r on its diagonal, and $\mathbf{C}_k$ $(R \times R)$

is the diagonal matrix with the kth elements of the vectors $\mathbf{c}_r$ on its diagonal. The model part of the CP model is characterized by $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{\Omega})$, where $\mathbf{C}$ $(d_3 \times R)$ has the vectors $\mathbf{c}_r$ as columns. We refer to $\mathbf{A}, \mathbf{B}, \mathbf{C}$ as the component matrices and to $\mathbf{\Omega}$ as the weights matrix.

The most attractive feature of CP is its uniqueness property. Kruskal [22] has shown that, for fixed residuals $\underline{\mathbf{E}}$, the vectors $\mathbf{a}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r$ and the weights ω_r are unique up to sign changes and a reordering of the summands in (1.5) if

$$k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geq 2\,R + 2\,, \tag{1.7}$$

where $k_{\mathbf{A}}$, $k_{\mathbf{B}}$, $k_{\mathbf{C}}$ denote the k-ranks of the component matrices. The k-rank of a matrix is the largest number x such that every subset of x columns of the matrix is linearly independent. Hence, contrary to the matrix principal components model, the CP components are rotationally unique if (1.7) holds.

However, the practical use of CP has been hampered by the occurrence of diverging CP components. In the majority of such cases, exactly two components displayed the following pattern. Let the model parameters of the nth update of a CP algorithm be denoted by a superscript (n). In the case of two diverging CP components, say s and t, the weights $\omega_s^{(n)}$ and $\omega_t^{(n)}$ become arbitrarily large in magnitude while the vectors $\mathbf{a}_s^{(n)}$, $\mathbf{b}_s^{(n)}$, and $\mathbf{c}_s^{(n)}$ become nearly identical (up to sign changes) to $\mathbf{a}_t^{(n)}$, $\mathbf{b}_t^{(n)}$, and $\mathbf{c}_t^{(n)}$ such that

$$\sum_{r=s,t} \omega_r^{(n)} \, (\mathbf{a}_r^{(n)} \circ \mathbf{b}_r^{(n)} \circ \mathbf{c}_r^{(n)})$$

remains "small." Hence, the contributions of components s and t diverge in nearly opposite directions, but their sum still contributes to a better fit of the CP model. The CP algorithm becomes very slow when this occurs; see Mitchell and Burdick [28]. When the CP algorithm is terminated, the CP components obtained are said to form a *degenerate CP solution*. Since this use of the term *degenerate* is different from its general meaning in mathematics, we will speak of *diverging CP components* instead. This also reflects the fact that this phenomenon occurs when running a CP algorithm, while a degenerate CP solution suggests a property of one CP solution only.

The first case of two diverging CP components was reported in Harshman and Lundy [14]. Contrived examples are given by ten Berge, Kiers, and De Leeuw [37] and Paatero [30]. The latter has also constructed sequences of CP updates with three and four diverging components.

Kruskal, Harshman, and Lundy [24] have argued that diverging CP components occur due to the fact that the array $\underline{\mathbf{X}}$ has no best rank-R approximation. They reason that every sequence of CP updates of which the objective value is approaching the infimum of the CP objective function must fail to converge and displays a pattern of diverging CP components. Stegeman [34] confirms this statement (under some regularity condition) for generic $p \times p \times 2$ arrays of rank $p+1$ with $R = p$. Stegeman [35] confirms the statement of [24] for generic $3 \times 3 \times p$ arrays with symmetric slices of rank $p+1$ with $R = p$, $p = 4, 5$, for generic $3 \times 3 \times 5$ arrays of rank 6 with $R = 5$, and for generic $8 \times 4 \times 3$ arrays of rank 9 with $R = 8$.

For given $d_1, d_2, d_3 \geq 2$, let

$$\mathcal{S}_R = \{\underline{\mathbf{Y}} \in \mathbb{R}^{d_1 \times d_2 \times d_3} : \operatorname{rank}_\circ(\underline{\mathbf{Y}}) \leq R\}\,, \tag{1.8}$$

and let $\overline{\mathcal{S}}_R$ denote its closure. We assume that $\operatorname{rank}_\circ(\underline{\mathbf{X}}) > R$. Hence, if $\underline{\mathbf{X}}$ has a best rank-R approximation, it will be a boundary point of $\mathcal{S}_R$. So far, all but one

mathematically analyzed case of diverging CP components (i.e., ten Berge, Kiers, and De Leeuw [37], Paatero [30], Stegeman [34, 35], and the results in the present paper) are due to the fact that the sequence $\underline{\mathbf{Y}}^{(n)} \in \mathcal{S}_R$ of CP updates converges to a boundary point $\widetilde{\underline{\mathbf{X}}}$ of $\mathcal{S}_R$ with $\text{rank}_\circ(\widetilde{\underline{\mathbf{X}}}) > R$, i.e., $\widetilde{\underline{\mathbf{X}}} \in \overline{\mathcal{S}}_R \backslash \mathcal{S}_R$, where $\widetilde{\underline{\mathbf{X}}}$ is a best approximation of $\underline{\mathbf{X}}$ from $\overline{\mathcal{S}}_R$. In these cases, the phenomenon of diverging CP components can be formalized as follows. There exist disjoint index sets $I_1, \ldots, I_m \subset \{1, \ldots, R\}$ such that as $\underline{\mathbf{Y}}^{(n)} \to \widetilde{\underline{\mathbf{X}}}$,

$$|\omega_r^{(n)}| \to \infty \qquad \text{for all } r \in I_j\,, \quad j = 1, \ldots, m\,, \tag{1.9}$$

$$\text{while} \quad \left\| \sum_{r \in I_j} \omega_r^{(n)} \, (\mathbf{a}_r^{(n)} \circ \mathbf{b}_r^{(n)} \circ \mathbf{c}_r^{(n)}) \right\| \qquad \text{is bounded}\,, \quad j = 1, \ldots, m\,. \tag{1.10}$$

For two diverging CP components, we have $m = 1$ and $\text{card}(I_1) = 2$. For three diverging CP components, we have $m = 1$ and $\text{card}(I_1) = 3$. For two groups of diverging CP components we have $m = 2$, et cetera. For the case of generic $p \times q \times 2$ arrays it will be shown in section 3 how the rank of $\widetilde{\underline{\mathbf{X}}}$ is related to the number of groups m and the number of diverging CP components in each group.

Note that we do not consider cases where $\text{rank}_\circ(\underline{\mathbf{X}}) = R$ and its CP decomposition resembles a case of diverging CP components, examples of which can be found in Mitchell and Burdick [28] and Paatero [30].

If $\underline{\mathbf{X}}$ does not have a best rank-R approximation, this implies that all best approximations $\widetilde{\underline{\mathbf{X}}}$ of $\underline{\mathbf{X}}$ from $\overline{\mathcal{S}}_R$ have at least rank $R+1$. In the cases analyzed so far, any sequence $\underline{\mathbf{Y}}^{(n)}$ of CP updates converging to $\widetilde{\underline{\mathbf{X}}}$ has been shown (under some regularity conditions) to exhibit diverging CP components in this situation. Hence, modified CP algorithms designed to avoid diverging CP components (e.g., Rayens and Mitchell [31] and Cao et al. [8]) are no remedy here.

As mentioned above, there is one known case where $\underline{\mathbf{X}}$ has a best rank-R approximation, but diverging CP components may still occur. This is the case of $3 \times 3 \times 5$ arrays of rank 6 and $R = 5$. Here, a best rank-5 approximation $\widetilde{\underline{\mathbf{X}}}$ of $\underline{\mathbf{X}}$ may exist while sequences $\underline{\mathbf{Y}}^{(n)}$ of CP updates converging to $\widetilde{\underline{\mathbf{X}}}$ sometimes show diverging CP components and sometimes do not; see Stegeman [35]. This is due to the partial uniqueness of the CP decomposition of $\underline{\mathbf{Y}}^{(n)}$; see ten Berge [40].

Diverging CP components are a problem in the analysis of three-way arrays, since the obtained CP solution is hardly interpretable. Diverging CP components can be avoided by imposing orthogonality constraints on the components matrices (see Harshman and Lundy [14]) but this will come with some loss of fit. Lim [27] shows that for nonnegative $\underline{\mathbf{X}}$ and nonnegative component matrices there always exists an optimal CP solution and diverging CP components do not occur.

1.4. Matrix pencils and the rank of $p \times q \times 2$ arrays. A matrix pencil $\mathbf{X}_1 + \lambda \mathbf{X}_2$ consists of two matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ with elements in a field $\mathcal{F}$ and a scalar λ. A matrix pencil is called *regular* if both $\mathbf{X}_1$ and $\mathbf{X}_2$ are square matrices and there exists an λ such that $\det(\mathbf{X}_1 + \lambda \mathbf{X}_2) \neq 0$. In all other cases, the pencil is called *singular*. For regular matrix pencils, equivalence results and a canonical form were established by Weierstrass [42]. The corresponding theory for singular pencils was developed by Kronecker [20]. For an overview of matrix pencil theory we refer the reader to Gantmacher [12, Chapter XII].

Ja' Ja' [19] has extended Kronecker's [20] equivalence results for $p \times q$ matrix pencils to $p \times q \times 2$ arrays. The same author [18] obtained results on the multiplicative complexity of computing two bilinear forms by considering the Kronecker canonical form of the associated matrix pencil and gave a complete characterization of the rank of the associated $p \times q \times 2$ array. In particular, Ja' Ja' [18] showed that for $p \geq q$,

$$\underline{\mathbf{X}} \in \mathbb{R}^{p \times q \times 2} \quad \Longrightarrow \quad \operatorname{rank}_\circ(\underline{\mathbf{X}}) \leq q + \min(q, \operatorname{floor}(p/2)) \,, \tag{1.11}$$

where the rank is over the real field and $\operatorname{floor}(x)$ denotes the largest integer smaller than or equal to x; see also Kruskal [23]. The upper bound (1.11) is sharp, i.e., there exist $p \times q \times 2$ arrays with rank equal to the upper bound. For later use, we state the following result, also due to Ja' Ja' [18].

PROPOSITION 1.2. *Let $\underline{\mathbf{X}} \in \mathbb{R}^{p \times p \times 2}$ with $p \times p$ slices $\mathbf{X}_i$, $i = 1, 2$. Suppose $det(\mathbf{X}_1) \neq 0$ and $\mathbf{X}_2\mathbf{X}_1^{-1}$ has p real eigenvalues. Let the Jordan normal form (see Gantmacher* [12, Chapter VI]) *of $\mathbf{X}_2\mathbf{X}_1^{-1}$ be given by* $\operatorname{diag}(J_{n_1}(\lambda_1), \ldots, J_{n_r}(\lambda_r))$, *where $J_{n_j}(\lambda_j)$ denotes an $n_j \times n_j$ Jordan block with diagonal elements equal to λ_j. Then*

$$\operatorname{rank}_\circ(\underline{\mathbf{X}}) = p + k \,, \tag{1.12}$$

where the rank is over the real field and k is the number of Jordan blocks $J_{n_j}(\lambda_j)$ with $n_j > 1$.

For a $p \times p$ matrix $\mathbf{Z}$ with eigenvalues $\lambda_1, \ldots, \lambda_r$, we define the *algebraic multiplicity* of λ_j as the multiplicity of λ_j as root of the characteristic polynomial $\det(\mathbf{Z} - \lambda \mathbf{I}_p)$, and the *geometric multiplicity* of λ_j as the maximum number of linearly independent eigenvectors of $\mathbf{Z}$ associated with λ_j (i.e., the dimensionality of the eigenspace of λ_j). Recall that for $\mathbf{Z} = \operatorname{diag}(J_{n_1}(\lambda_1), \ldots, J_{n_r}(\lambda_r))$, the eigenvalues are $\lambda_1, \ldots, \lambda_r$ (not necessarily distinct), and each Jordan block $J_{n_j}(\lambda_j)$ adds n_j to the algebraic multiplicity of λ_j and 1 to the geometric multiplicity of λ_j. This establishes a relation between the eigenvalues of $\mathbf{X}_2\mathbf{X}_1^{-1}$ and the rank of the array $\underline{\mathbf{X}}$ in Proposition 1.2. In particular, if $\mathbf{X}_2\mathbf{X}_1^{-1}$ has p real eigenvalues and is diagonalizable, then $\operatorname{rank}_\circ(\underline{\mathbf{X}}) = p$ (see also Ten Berge [38]). Ja' Ja' [18] also showed that if $\mathbf{X}_2\mathbf{X}_1^{-1}$ has at least one pair of complex eigenvalues, then $\operatorname{rank}_\circ(\underline{\mathbf{X}}) \geq p + 1$ (see also [38]).

For generic $p \times q \times 2$ arrays, Ten Berge and Kiers [39] showed that, for $p > q$, the rank of $\underline{\mathbf{X}}$ is equal to $\min(p, 2q)$ almost everywhere, i.e., $\operatorname{rank}(\underline{\mathbf{X}}) \neq \min(p, 2q)$ on a set of zero volume in $\mathbb{R}^{p \times q \times 2}$. We call this rank value the *typical rank*. The same authors show that for $p = q$, the typical rank of $\underline{\mathbf{X}}$ is two-valued, namely $\{p,\ p+1\}$, where the sets of both rank values have positive volume. Notice that for $p \times q \times 2$ arrays the set $\mathcal{S}_R$ of arrays with rank less than or equal to R has dimensionality $2pq$ if R is larger than or equal to the typical rank. If R is smaller than the typical rank, then $\mathcal{S}_R$ has dimensionality lower than $2pq$. Analogously, if R is larger than the typical rank, then the set $\mathcal{S}_R^c = \mathbb{R}^{p \times q \times 2} \backslash \mathcal{S}_R$ has dimensionality lower than $2pq$.

Notice that if $p \geq 2q$, then both the typical rank and the maximum rank (1.11) are equal to $2q$. If $2q > p > q$, then the typical rank equals p, while the maximum rank equals

$$q + \operatorname{floor}(p/2) \geq ((p+1)/2) + \operatorname{floor}(p/2) \geq p \,. \tag{1.13}$$

Bini [4] has studied the border rank of so-called *nondegenerate* $p \times q \times 2$ arrays, where a 3-way array $\underline{\mathbf{X}} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ is called nondegenerate if $\operatorname{rank}_i(\underline{\mathbf{X}}) = d_i$ for $i = 1, 2, 3$. The use of the term *nondegenerate* here is a bit strange, since generic $p \times q \times 2$ arrays

are nondegenerate only if $p \leq 2q$ and $q \leq 2p$. The following result is based on the results of Ja' Ja' [18] and is due to Bini [4].

PROPOSITION 1.3. *Let* $\underline{\mathbf{X}} \in \mathbb{R}^{p\times q\times 2}$ *such that* $rank_1(\underline{\mathbf{X}}) = p$, $rank_2(\underline{\mathbf{X}}) = q$, *and* $rank_3(\underline{\mathbf{X}}) = 2$. *Let* $\mathbf{X}_1$ *and* $\mathbf{X}_2$ *be the* $p \times q$ *slices of* $\underline{\mathbf{X}}$.

(i) *Let* $p = q$ *and* $det(\mathbf{X}_1) \neq 0$. *Then* $rank_B(\underline{\mathbf{X}}) = p$ *if and only if* $\mathbf{X}_2\mathbf{X}_1^{-1}$ *has* p *real eigenvalues. If* $\mathbf{X}_2\mathbf{X}_1^{-1}$ *has at least one pair of complex eigenvalues, then* $rank_B(\underline{\mathbf{X}}) = p + 1$.

(ii) *Let* $p = q$ *and* $det(\mathbf{X}_1) = 0$. *Then* $rank_B(\underline{\mathbf{X}}) = p$ *if* $det(\mathbf{X}_1 + \lambda\mathbf{X}_2)$ *has* p *real roots* λ. *If* $det(\mathbf{X}_1 + \lambda\mathbf{X}_2)$ *has at least one pair of complex roots, then* $rank_B(\underline{\mathbf{X}}) \in \{p, p + 1\}$.

(iii) *If* $p > q$, *then* $rank_B(\underline{\mathbf{X}}) = p$ *if* $det(\mathbf{Y}_1 + \lambda\mathbf{Y}_2)$ *has only real roots* λ, *where* $\mathbf{Y}_1 + \lambda\mathbf{Y}_2$ *is the regular pencil kernel in the Kronecker canonical form of* $\mathbf{X}_1 + \lambda\mathbf{X}_2$. *If* $det(\mathbf{Y}_1 + \lambda\mathbf{Y}_2)$ *has at least one pair of complex roots, then* $rank_B(\underline{\mathbf{X}}) \in \{p, p + 1\}$.

Next, we discuss the link between the results in the present paper and the existing results mentioned above. In this paper, we consider the low-rank approximation of generic real-valued $p \times q \times 2$ arrays, where we assume $p \geq q$ without loss of generality. Such an array has typical rank $\min(p, 2q)$ and we show whether or not it has a best rank-R approximation, with $R < \min(p, 2q)$. If such a generic array is nondegenerate in the sense of Bini [4], then it has typical rank $\min(p, 2q) = p$ and $\text{rank}_\circ(\underline{\mathbf{X}}) \geq p$; see (1.4). If $\text{rank}_\circ(\underline{\mathbf{X}}) = p$, then it follows from (1.4) that also $\text{rank}_B(\underline{\mathbf{X}}) = p$. The case $\text{rank}_B(\underline{\mathbf{X}}) = p+1$ is possible only for arrays $\underline{\mathbf{X}}$ with rank larger than the typical rank p, and such arrays are not generic. Hence, our results are not covered by Proposition 1.3.

To obtain our results, we study the boundary of $\mathcal{S}_R$, the set of $p \times q \times 2$ arrays with rank at most R, and distinguish boundary arrays lying in $\mathcal{S}_R$ from those with rank larger than R. Clearly, a boundary array $\underline{\widetilde{\mathbf{X}}}$ of $\mathcal{S}_R$ has border rank at most R and is not a nondegenerate array in the sense of Bini [4] since $\text{rank}_1(\underline{\widetilde{\mathbf{X}}}) \leq R < p$; see (1.4). However, we transform the set $\mathcal{S}_R$ to a subset of the smaller space of $R \times R \times 2$ arrays if $R \leq q$, or $R \times q \times 2$ arrays if $R > q$, and Propositions 1.2 and 1.3 apply to individual boundary arrays in this smaller space. But to answer the question whether a generic $p \times q \times 2$ array has a best rank-R approximation almost everywhere, or on a set of positive volume, or on a set of zero volume, we need dimensionality arguments to establish which rank values have positive volume on the boundary of $\mathcal{S}_R$. Moreover, we do not use the Kronecker canonical form in (iii) of Proposition 1.3 to obtain our results.

This paper is organized as follows. We present our results on the existence of a best rank-R approximation to generic $p \times q \times 2$ arrays in section 2. In section 3 we show (under some regularity condition) that if there is no best rank-R approximation, then, in the CP model, any sequence $\underline{\mathbf{Y}}^{(n)} \in \mathcal{S}_R$ converging to an optimal boundary point $\underline{\widetilde{\mathbf{X}}}$ of $\mathcal{S}_R$ (with $\text{rank}_\circ(\underline{\widetilde{\mathbf{X}}}) > R$) will exhibit diverging CP components as defined by (1.9) and (1.10). Moreover, we show that there is a direct relation between the rank of $\underline{\widetilde{\mathbf{X}}}$ and the number of groups of diverging CP components. This extends Stegeman [34] who considered rank-p approximations to generic $p \times p \times 2$ arrays of rank $p + 1$. In section 4 we illustrate our results by means of calculating rank-R approximations to random $p \times q \times 2$ arrays for a variety of values for p, q, and R. Finally, section 5 contains a discussion on the presented results.

2. Low-rank approximation of generic $p \times q \times 2$ arrays. For generic $\underline{\mathbf{X}} \in \mathbb{R}^{p\times q\times 2}$, we consider the problem

$$\begin{aligned} &\text{Minimize} \|\underline{\mathbf{X}} - \underline{\mathbf{Y}}\| \\ &\text{subject to } \underline{\mathbf{Y}} \in \mathcal{S}_R\,, \end{aligned} \tag{2.1}$$

where $\mathcal{S}_R$ is the set of real-valued $p \times q \times 2$ arrays of rank at most R (see (1.8)) and the rank is taken over the real field. We also consider the related problem

$$\begin{aligned} &\text{Minimize} \|\underline{\mathbf{X}} - \underline{\mathbf{Y}}\| \\ &\text{subject to } \underline{\mathbf{Y}} \in \overline{\mathcal{S}}_R\,, \end{aligned} \tag{2.2}$$

where $\overline{\mathcal{S}}_R$ is the closure of $\mathcal{S}_R$, i.e., the union of $\mathcal{S}_R$ and all its boundary points. Suppose $\text{rank}_\circ(\underline{\mathbf{X}}) > R$. There holds that any optimal solution of problem (2.2) is a boundary point of $\mathcal{S}_R$. Indeed, from any interior point $\underline{\mathbf{Y}}$ of $\mathcal{S}_R$ a line to $\underline{\mathbf{X}}$ can be drawn which intersects with the boundary of $\mathcal{S}_R$. Suppose the intersection occurs at boundary point $\widetilde{\underline{\mathbf{X}}}$. Then $\widetilde{\underline{\mathbf{X}}}$ has a lower objective value (i.e., is closer to $\underline{\mathbf{X}}$) than the interior point $\underline{\mathbf{Y}}$. Hence, problem (2.1) has an optimal solution, or, equivalently, $\underline{\mathbf{X}}$ has a best rank-R approximation, if there exists a boundary point $\widetilde{\underline{\mathbf{X}}} \in \mathcal{S}_R$ which is an optimal solution of problem (2.2). Clearly, this always holds if $\mathcal{S}_R$ is a closed set. However, De Silva and Lim [10] have shown that $\mathcal{S}_R$ is closed only for $R = 1$. Hence, an investigation of the boundary points of $\mathcal{S}_R$ is necessary to ascertain whether a best rank-R approximation exists almost everywhere, or on a set of positive volume, or on a set of zero volume.

We consider all possible combinations of p, q, and R, where, without loss of generality, we assume that $p \geq q$. As mentioned in section 1, we transform the set $\mathcal{S}_R$ to a subset of the smaller space of $R \times R \times 2$ arrays if $R \leq q$ or $R \times q \times 2$ arrays if $R > q$ and use the results of Stegeman [34] and Proposition 1.2 to characterize the boundary points of $\mathcal{S}_R$; in particular, whether they are in $\mathcal{S}_R$ or in $\mathcal{S}_R^c$.

Unfortunately, our results are not complete. That is, they rely on conjectures relating the dimensionality of parts of the boundary of $\mathcal{S}_R$ to the existence of optimal solutions of problem (2.1). Table 2.1 gives a summary of our results. Below, we consider all cases in Table 2.1 and state explicitly whether we use a conjecture. Except for cases 1, 4, and 6, the statements on the existence of a best rank-R approximation are (partly) based on conjectures.

Cases 1, 4, and 6. In these cases, R is larger than or equal to the typical rank of $\underline{\mathbf{X}}$, i.e., $\underline{\mathbf{X}}$ itself lies in $\mathcal{S}_R$ almost everywhere. Hence, the best rank-R approximation of $\underline{\mathbf{X}}$ is $\underline{\mathbf{X}}$ itself.

Cases 2, 3, 5, 7, 8, and 9. For these cases, there holds $\underline{\mathbf{X}} \notin \mathcal{S}_R$ almost everywhere. As discussed above, we need to characterize the boundary of $\mathcal{S}_R$. We define the following subsets of $\mathbb{R}^{p \times q \times 2}$. Let

$$\mathcal{W}_R = \{\underline{\mathbf{Y}} \in \mathbb{R}^{p \times q \times 2} : \text{rank}[\mathbf{Y}_1 | \mathbf{Y}_2] \leq R\} = \{\underline{\mathbf{Y}} \in \mathbb{R}^{p \times q \times 2} : \text{rank}_1(\underline{\mathbf{Y}}) \leq R\}\,, \tag{2.3}$$

where $\mathbf{Y}_1$ and $\mathbf{Y}_2$ denote the $p \times q$ slices of $\underline{\mathbf{Y}}$, and let

$$\mathcal{V}_{R+1} = \{\underline{\mathbf{Y}} \in \mathcal{W}_R : \text{rank}_\circ(\underline{\mathbf{Y}}) \geq R + 1\} = \mathcal{W}_R \cap \mathcal{S}_R^c\,. \tag{2.4}$$

We need the following lemma.

LEMMA 2.1. *We have the following results.*

(i) *The set $\mathcal{W}_R$ is closed and the boundary of $\mathcal{W}_R$ is the set $\mathcal{W}_R$ itself.*

TABLE 2.1
Results (cases 1, 4, and 6) and conjectures (cases 2, 3, 5, 7, 8, and 9) on the existence of a best rank-R approximation to generic $p \times q \times 2$ arrays. Here, $p \geq q \geq 2$ and $R \geq 2$.

Case	$\underline{\mathbf{X}} \in \mathbb{R}^{p \times q \times 2}$	$\text{rank}_\circ(\underline{\mathbf{X}})$	R	Best rank-R approx. exists?
1	$p = q$	$p+1$	$R \geq p+1$	almost everywhere
2	$p = q$	$p+1$	$R = p$	zero volume
3	$p = q$	$p+1$	$R < p$	positive volume
4	$p = q$	p	$R \geq p$	almost everywhere
5	$p = q$	p	$R < p$	positive volume
6	$p > q$	$\min(p, 2q)$	$R \geq \min(p, 2q)$	almost everywhere
7	$p > q$	$\min(p, 2q)$	$\min(p, 2q) > R > q$	almost everywhere
8	$p > q$	$\min(p, 2q)$	$R = q$	positive volume
9	$p > q$	$\min(p, 2q)$	$R < q$	positive volume

(ii) *There holds*

$$\mathcal{W}_R = \mathcal{S}_R \cup \mathcal{V}_{R+1} \qquad \text{and} \qquad \mathcal{S}_R \cap \mathcal{V}_{R+1} = \emptyset\,. \tag{2.5}$$

(iii) *In case 2, there holds $\mathcal{W}_R = \mathbb{R}^{p \times q \times 2}$.*

Proof. Statement (i) follows from the fact that any matrix of rank at most R can be approximated arbitrarily well by rank-$(R+1)$ matrices.

Next, we prove (ii). From (1.4) it follows that $\mathcal{S}_R \subseteq \mathcal{W}_R$. Since $\mathcal{V}_{R+1} = \mathcal{W}_R \cap \mathcal{S}_R^c$, we have $\mathcal{S}_R \cap \mathcal{V}_{R+1} = \emptyset$ by definition. Hence, $\mathcal{W}_R = (\mathcal{W}_R \cap \mathcal{S}_R) \cup (\mathcal{W}_R \cap \mathcal{S}_R^c) = \mathcal{S}_R \cup \mathcal{V}_{R+1}$.

In case 2, we have $R = p = q$, which implies that $[\mathbf{Y}_1 | \, \mathbf{Y}_2]$ is a matrix of order $p \times 2p$. Obviously, this matrix always has rank less than or equal to p. Hence, $\mathcal{W}_R = \mathbb{R}^{p \times q \times 2}$ in case 2. This proves (iii). ∎

Next, we consider the boundary points of $\mathcal{S}_R$. The complement of $\mathcal{S}_R$ is equal to

$$\mathcal{S}_R^c = \mathcal{W}_R^c \cup \mathcal{V}_{R+1} \qquad \text{with} \qquad \mathcal{W}_R^c \cap \mathcal{V}_{R+1} = \emptyset\,. \tag{2.6}$$

Hence, the boundary of $\mathcal{S}_R$ consists of the boundary between $\mathcal{S}_R$ and $\mathcal{V}_{R+1}$ and the boundary between $\mathcal{S}_R$ and $\mathcal{W}_R^c$. Note that these two boundaries may have a nonempty intersection. We denote the boundary of $\mathcal{S}_R$ as $\partial\mathcal{S}_R$ and partition it into the following two sets. Let

$$\mathcal{U}_R^{(1)} = \partial\mathcal{S}_R \cap \partial\mathcal{V}_{R+1} \qquad \text{and} \qquad \mathcal{U}_R^{(2)} = \partial\mathcal{S}_R \cap (\partial\mathcal{V}_{R+1})^c\,. \tag{2.7}$$

Hence, $\mathcal{U}_R^{(1)}$ consists of all points on the boundary between $\mathcal{S}_R$ and $\mathcal{V}_{R+1}$, i.e., all points which can be approximated arbitrarily well from both $\mathcal{S}_R$ and $\mathcal{V}_{R+1}$.

From Lemma 2.1 it follows that

$$\partial\mathcal{S}_R \subseteq \mathcal{W}_R = \mathcal{S}_R \cup \mathcal{V}_{R+1}\,. \tag{2.8}$$

The following lemma states that $\mathcal{U}_R^{(2)}$ is either the empty set or is a subset of $\mathcal{S}_R$.

LEMMA 2.2. *We have the following results.*

(i) *In case 2, there holds $\mathcal{U}_R^{(2)} = \emptyset$.*

(ii) *In cases* 3, 5, 7, 8, *and* 9, *there holds* $\mathcal{U}_R^{(2)} \subseteq \mathcal{S}_R$.

Proof. From (2.8) it follows that $\mathcal{U}_R^{(1)} \subseteq \mathcal{W}_R$ and $\mathcal{U}_R^{(2)} \subseteq \mathcal{W}_R$. In case 2, we have $\mathcal{W}_R = \mathbb{R}^{p\times q\times 2}$ (see Lemma 2.1) and, hence, $\mathcal{V}_{R+1} = \mathcal{S}_R^c$. This implies that $\mathcal{U}_R^{(1)}$ consists of all points which can be approximated arbitrarily well from $\mathcal{S}_R$ and $\mathcal{S}_R^c$. Since these are all boundary points of $\mathcal{S}_R$, it follows that $\mathcal{U}_R^{(2)} = \emptyset$. This proves (i).

Next, consider cases 3, 5, 7, 8, and 9. Here, $\mathcal{W}_R^c \neq \emptyset$. Suppose $\underline{\mathbf{Y}} \in \mathcal{V}_{R+1}$ and $\underline{\mathbf{Y}} \in \partial\mathcal{S}_R$. Since $\mathcal{S}_R \cap \mathcal{V}_{R+1} = \emptyset$, it follows that $\underline{\mathbf{Y}} \in \partial\mathcal{V}_{R+1}$ and, hence, $\underline{\mathbf{Y}} \in \mathcal{U}_R^{(1)}$. This implies $\mathcal{V}_{R+1} \cap \mathcal{U}_R^{(2)} = \emptyset$. Moreover, from $\mathcal{U}_R^{(2)} \subseteq \mathcal{W}_R = \mathcal{S}_R \cup \mathcal{V}_{R+1}$, we obtain $\mathcal{U}_R^{(2)} \subseteq \mathcal{S}_R$. This proves (ii). ∎

In the remaining part of this section, we consider the part $\mathcal{U}_R^{(1)}$ of the boundary of $\mathcal{S}_R$. For each of the cases 2, 3, 5, 7, 8 and 9 in Table 2.1, we argue that the nonexistence of a best rank-R approximation is due to the fact that $\mathcal{U}_R^{(1)} \not\subseteq \mathcal{S}_R$.

Case 2. We have $p = q = R$, $\mathcal{W}_R = \mathbb{R}^{p\times q\times 2} = \mathcal{S}_R \cup \mathcal{V}_{R+1}$, $\mathcal{V}_{R+1} = \mathcal{S}_R^c$, $\mathcal{U}_R^{(2)} = \emptyset$, and $\underline{\mathbf{X}} \in \mathcal{S}_R^c$. The typical rank of $p \times p \times 2$ arrays is equal to $\{p,\, p+1\}$, where the sets of both rank values have positive volume. This implies that the sets $\mathcal{S}_R$ and $\mathcal{V}_{R+1}$ have equal dimensionality $2p^2$. We partition $\partial\mathcal{S}_R = \mathcal{U}_R^{(1)}$ into the following three sets. Let

$$\mathcal{U}_R^{(11)} = \{\underline{\mathbf{Y}} \in \mathcal{U}_R^{(1)} \text{ with } \mathbf{Y}_1 \text{ nonsingular and } \mathbf{Y}_2\mathbf{Y}_1^{-1} \text{ diagonalizable}\}\,, \tag{2.9}$$

$$\mathcal{U}_R^{(12)} = \{\underline{\mathbf{Y}} \in \mathcal{U}_R^{(1)} \text{ with } \mathbf{Y}_1 \text{ nonsingular and } \mathbf{Y}_2\mathbf{Y}_1^{-1} \text{ not diagonalizable}\}\,, \tag{2.10}$$

$$\mathcal{U}_R^{(13)} = \{\underline{\mathbf{Y}} \in \mathcal{U}_R^{(1)} \text{ with } \mathbf{Y}_1 \text{ singular}\}\,. \tag{2.11}$$

In Stegeman [34], it is shown that $\underline{\mathbf{Y}} \in \mathcal{U}_R^{(11)} \cup \mathcal{U}_R^{(12)}$ if and only if $\mathbf{Y}_2\mathbf{Y}_1^{-1}$ has p real eigenvalues which are not all distinct. Although its proof is entirely different, this result is closely related to Propositions 1.2 and 1.3(i). From Proposition 1.2 it follows that $\mathcal{U}_R^{(11)} \subset \mathcal{S}_R$ and $\mathcal{U}_R^{(12)} \subset \mathcal{S}_R^c$. Stegeman [34] also shows that the dimensionality of $\mathcal{U}_R^{(11)}$ is lower that the dimensionality of $\mathcal{U}_R^{(12)}$.

Any array in $\mathcal{U}_R^{(13)}$ can be approximated arbitrarily well by arrays in $\mathcal{U}_R^{(11)} \cup \mathcal{U}_R^{(12)}$. The reverse, however, is not true. This implies that $\mathcal{U}_R^{(13)}$ has lower dimensionality than $\mathcal{U}_R^{(11)} \cup \mathcal{U}_R^{(12)}$. Combined with the reasoning above, this implies that the subset of the boundary of $\mathcal{S}_R$ with the highest dimensionality is $\mathcal{U}_R^{(12)}$. Since these boundary points have rank larger than R, we conjecture that for a generic $\underline{\mathbf{X}} \in \mathbb{R}^{p\times p\times 2}$ array an optimal solution $\underline{\widetilde{\mathbf{X}}}$ of problem (2.2) has rank larger than R almost everywhere. Hence, we conjecture that problem (2.1) does not have an optimal solution almost everywhere, and $\underline{\mathbf{X}}$ does not have a best rank-R approximation almost everywhere.

Case 7. We have $p > q$ and $\min(p, 2q) > R > q$. From the definition of $\mathcal{W}_R$ in (2.3) it follows that $\underline{\mathbf{Y}} \in \mathcal{W}_R$ if and only if there exists a nonsingular matrix $\mathbf{S}$ such that

$$\mathbf{S}\,\mathbf{Y}_1 = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{O} \end{bmatrix} \qquad \text{and} \qquad \mathbf{S}\,\mathbf{Y}_2 = \begin{bmatrix} \mathbf{H}_2 \\ \mathbf{O} \end{bmatrix}, \tag{2.12}$$

where $\mathbf{H}_1$ and $\mathbf{H}_2$ are $R \times q$ matrices and $\mathbf{O}$ is the $(p-R) \times q$ all-zero matrix. By $\underline{\mathbf{H}}$ we denote the $R \times q \times 2$ array with slices $\mathbf{H}_1$ and $\mathbf{H}_2$. The transformation $\mathbf{S}$ in (2.12)

is rank preserving, i.e., $\text{rank}_\circ(\underline{\mathbf{Y}}) = \text{rank}_\circ(\underline{\mathbf{H}})$. Hence, it follows from (2.4) and (2.5) that

$$\mathcal{S}_R = \{\underline{\mathbf{Y}} \in \mathcal{W}_R : \ \text{rank}_\circ(\underline{\mathbf{H}}) \leq R \ \text{ for } \underline{\mathbf{H}} \text{ in (2.12)}\} \tag{2.13}$$

and

$$\mathcal{V}_{R+1} = \{\underline{\mathbf{Y}} \in \mathcal{W}_R : \ \text{rank}_\circ(\underline{\mathbf{H}}) \geq R+1 \ \text{ for } \underline{\mathbf{H}} \text{ in (2.12)}\}\,. \tag{2.14}$$

It is important to note that the set of arrays $\underline{\mathbf{H}}$ which can be obtained in (2.12) has full dimensionality $2qR$. The typical rank of $R \times q \times 2$ arrays (with $2q > R > q$) equals R. When the maximal rank of $R \times q \times 2$ arrays equals their typical rank, we have $\mathcal{V}_{R+1} = \emptyset$, and $\mathcal{S}_R = \mathcal{W}_R$ is a closed set. This is the case when $R = 4$ and $q = 3$, e.g., see (1.11). Since $\mathcal{S}_R$ is closed, a best rank-R approximation to $\underline{\mathbf{X}}$ exists almost everywhere.

Next, suppose the maximal rank of $R \times q \times 2$ arrays is larger than their typical rank R, i.e., $\mathcal{V}_{R+1} \neq \emptyset$. This is the case when $R = 5$ and $q = 3$, e.g., see (1.11). For the arrays in $\mathcal{V}_{R+1}$, the array $\underline{\mathbf{H}}$ has a *nontypical* rank value, i.e., larger than or equal to $R+1$. This implies that $\mathcal{V}_{R+1}$ has lower dimensionality than the set $\mathcal{S}_R$. From (2.4) and (2.5) it then follows that any $\underline{\mathbf{Y}} \in \mathcal{V}_{R+1}$ can be approximated arbitrarily closely by arrays in $\mathcal{S}_R$ and it holds that $\mathcal{U}_R^{(1)} = \mathcal{V}_{R+1}$; see (2.7). Moreover, any array in $\mathcal{S}_R$ can be approximated arbitrarily well from $\mathcal{W}_R^{\text{c}}$ and, hence, $\mathcal{U}_R^{(2)} = \mathcal{S}_R$. This implies that

$$\partial\mathcal{S}_R = \mathcal{V}_{R+1} \cup \mathcal{S}_R = \mathcal{W}_R = \overline{\mathcal{S}}_R\,. \tag{2.15}$$

Since $\mathcal{V}_{R+1}$ has lower dimensionality than $\mathcal{S}_R$, we conjecture that an optimal solution $\widetilde{\underline{\mathbf{X}}}$ of problem (2.2) lies in $\mathcal{S}_R$ almost everywhere. Hence, we conjecture that $\underline{\mathbf{X}}$ has a best rank-R approximation almost everywhere.

Case 8. We have $p > q$ and $R = q$. As in case 7, there holds $\underline{\mathbf{Y}} \in \mathcal{W}_R$ if and only if a nonsingular $\mathbf{S}$ exists such that (2.12) holds. The sets $\mathcal{S}_R$ and $\mathcal{V}_{R+1}$ are defined by (2.13) and (2.14), respectively. The array $\underline{\mathbf{H}}$ in (2.12) has order $R \times R \times 2$ and the typical rank of $R \times R \times 2$ arrays is equal to $\{R,\, R+1\}$, where the sets of both rank values have positive volume. As in case 2, this implies that the sets $\mathcal{S}_R$ and $\mathcal{V}_{R+1}$ have equal dimensionality.

Let $\underline{\mathbf{X}}^*$ be an optimal solution of the following problem:

$$\begin{aligned} &\text{Minimize} \|\underline{\mathbf{X}} - \underline{\mathbf{Y}}\| \\ &\text{subject to } \underline{\mathbf{Y}} \in \mathcal{W}_R\,. \end{aligned} \tag{2.16}$$

Based on Lemma 2.1, we conjecture that, for generic $\underline{\mathbf{X}}$, the set where $\underline{\mathbf{X}}^* \in \mathcal{S}_R$ and the set where $\underline{\mathbf{X}}^* \in \mathcal{V}_{R+1}$ both have positive volume. If $\underline{\mathbf{X}}^* \in \mathcal{S}_R$, then $\underline{\mathbf{X}}^*$ is an optimal solution of the problem (2.1) and, hence, $\underline{\mathbf{X}}$ has a best rank-R approximation. If all optimal solutions of problem (2.16) lie in $\mathcal{V}_{R+1}$, then $\underline{\mathbf{X}}$ may not have a best rank-R approximation. This will be explained below.

The boundary of $\mathcal{S}_R$ is partitioned into $\mathcal{U}_R^{(1)}$ and $\mathcal{U}_R^{(2)}$, where $\mathcal{U}_R^{(2)} \subseteq \mathcal{S}_R$; see (2.7) and Lemma 2.2. Analogous to case 2, we partition $\mathcal{U}_R^{(1)}$ into the following sets:

$$\mathcal{U}_R^{(11)} = \{\underline{\mathbf{Y}} \in \mathcal{U}_R^{(1)} \text{ with } \mathbf{H}_1 \text{ in (2.12) nonsingular and } \mathbf{H}_2\mathbf{H}_1^{-1} \text{ diagonalizable}\}\,, \tag{2.17}$$

$$\mathcal{U}_R^{(12)} = \{\underline{\mathbf{Y}} \in \mathcal{U}_R^{(1)} \text{ with } \mathbf{H}_1 \text{ in (2.12) nonsingular and } \mathbf{H}_2\mathbf{H}_1^{-1} \text{ not diagonalizable}\}, \tag{2.18}$$

$$\mathcal{U}_R^{(13)} = \{\underline{\mathbf{Y}} \in \mathcal{U}_R^{(1)} \text{ with } \mathbf{H}_1 \text{ in (2.12) singular}\}. \tag{2.19}$$

Note that since $\operatorname{rank}(\mathbf{H}_1) = \operatorname{rank}(\mathbf{S}\,\mathbf{Y}_1) = \operatorname{rank}(\mathbf{Y}_1)$ and $\operatorname{rank}_\circ(\underline{\mathbf{Y}}) = \operatorname{rank}_\circ(\mathbf{H})$, combined with Proposition 1.2, it follows that the sets in (**??**)–(2.19) do not depend on the choice of $\mathbf{S}$ in (2.12). Analogous to case 2, there holds $\underline{\mathbf{Y}} \in \mathcal{U}_R^{(11)} \cup \mathcal{U}_R^{(12)}$ if and only if $\mathbf{H}_2\mathbf{H}_1^{-1}$ has R real eigenvalues which are not all distinct. Also, we have $\mathcal{U}_R^{(11)} \subset \mathcal{S}_R$ and $\mathcal{U}_R^{(12)} \subset \mathcal{S}_R^c$. Moreover, the sets $\mathcal{U}_R^{(11)}$ and $\mathcal{U}_R^{(13)}$ have lower dimensionality than $\mathcal{U}_R^{(12)}$. Hence, if all optimal solutions of problem (2.2) lie in $\mathcal{U}_R^{(1)}$, then they have rank larger than R almost everywhere on $\mathcal{U}_R^{(1)}$, and $\underline{\mathbf{X}}$ does not have a best rank-R approximation.

Above, we conjectured that, for generic $\underline{\mathbf{X}}$, an optimal solution $\underline{\mathbf{X}}^*$ of problem (2.16) lies in $\mathcal{S}_R$ on a set of positive volume. Hence, we conjecture that $\underline{\mathbf{X}}$ has a best rank-R approximation on a set of positive volume. Next, we argue that the set on which $\underline{\mathbf{X}}$ has no best rank-R approximation also has positive volume. This can be seen as follows. Let $\mathbf{Y} \in \mathcal{V}_{R+1}$ be an interior point of $\mathcal{V}_{R+1}$ on $\mathcal{W}_R$, i.e., for a small $\epsilon > 0$ we have $\mathcal{B}_\epsilon(\underline{\mathbf{Y}}) = \{\underline{\mathbf{Z}} \in \mathcal{W}_R : \|\underline{\mathbf{Z}} - \underline{\mathbf{Y}}\| < \epsilon\} \subset \mathcal{V}_{R+1}$. This is the case if $\mathbf{H}_2(\mathbf{H}_1)^{-1}$ has R distinct eigenvalues which are not all real, where $\mathbf{H}_i$ are as in (2.12); see Stegeman [34]. For any interior point $\underline{\mathbf{Y}}$ of $\mathcal{V}_{R+1}$, we can find a set $\mathcal{D} \subset \mathcal{W}_R^c$ close to $\mathcal{B}_\epsilon(\underline{\mathbf{Y}})$ such that $\mathcal{D}$ has positive volume and for any $\underline{\mathbf{X}} \in \mathcal{D}$ problem (2.16) will have all optimal solutions in $\mathcal{B}_\epsilon(\underline{\mathbf{Y}})$. Moreover, for $\underline{\mathbf{Y}}$ close enough to some point on $\mathcal{U}_R^{(12)}$, i.e., the boundary between $\mathcal{S}_R$ and $\mathcal{V}_{R+1}$, we conjecture that all optimal solutions of problem (2.2) will lie on the boundary $\mathcal{U}_R^{(12)}$ and have rank larger than R. Hence, we conjecture that no $\underline{\mathbf{X}} \in \mathcal{D}$ has a best rank-R approximation, where $\mathcal{D}$ has positive volume. Therefore, we conjecture that the set on which $\underline{\mathbf{X}}$ has no best rank-R approximation has positive volume.

Cases 3, 5, and 9. We have $p \geq q$ and $R < q$. Instead of $\mathcal{W}_R$, we define

$$\widetilde{\mathcal{W}}_R = \left\{\underline{\mathbf{Y}} \in \mathbb{R}^{p\times q\times 2} : \operatorname{rank}[\mathbf{Y}_1|\,\mathbf{Y}_2] \leq R \text{ and } \operatorname{rank}\begin{bmatrix}\mathbf{Y}_1\\ \mathbf{Y}_2\end{bmatrix} \leq R\right\}$$

$$= \{\underline{\mathbf{Y}} \in \mathbb{R}^{p\times q\times 2} : \operatorname{rank}_1(\underline{\mathbf{Y}}) \leq R \text{ and } \operatorname{rank}_2(\underline{\mathbf{Y}}) \leq R\}. \tag{2.20}$$

Analogous to Lemma 2.1, the set $\widetilde{\mathcal{W}}_R$ is closed. It can be seen that $\underline{\mathbf{Y}} \in \widetilde{\mathcal{W}}_R$ if and only if there exist nonsingular $\mathbf{S}$ and $\mathbf{T}$ such that

$$\mathbf{S}\,\mathbf{Y}_1\,\mathbf{T} = \begin{bmatrix}\mathbf{G}_1 & \mathbf{O}\\ \mathbf{O} & \mathbf{O}\end{bmatrix} \quad \text{and} \quad \mathbf{S}\,\mathbf{Y}_2\,\mathbf{T} = \begin{bmatrix}\mathbf{G}_2 & \mathbf{O}\\ \mathbf{O} & \mathbf{O}\end{bmatrix}, \tag{2.21}$$

where $\mathbf{G}_1$ and $\mathbf{G}_2$ are $R \times R$ matrices. We denote the $R \times R \times 2$ array with slices $\mathbf{G}_1$ and $\mathbf{G}_2$ by $\underline{\mathbf{G}}$. Note that the set of arrays $\underline{\mathbf{G}}$ which can be obtained by (2.21) has full dimensionality $2R^2$. Since the transformations $\mathbf{S}$ and $\mathbf{T}$ are rank preserving, we have, analogous to (2.13), that

$$\mathcal{S}_R = \{\underline{\mathbf{Y}} \in \widetilde{\mathcal{W}}_R : \operatorname{rank}_\circ(\underline{\mathbf{G}}) \leq R \text{ for } \underline{\mathbf{G}} \text{ in (2.21)}\}. \tag{2.22}$$

We define

$$\widetilde{\mathcal{V}}_{R+1} = \{\underline{\mathbf{Y}} \in \widetilde{\mathcal{W}}_R : \ \text{rank}_\circ(\underline{\mathbf{G}}) \geq R+1 \ \text{ for } \underline{\mathbf{G}} \text{ in (2.21)}\}\,. \tag{2.23}$$

Analogous to Lemma 2.1, there holds

$$\widetilde{\mathcal{W}}_R = \mathcal{S}_R \cup \widetilde{\mathcal{V}}_{R+1} \qquad \text{and} \qquad \mathcal{S}_R \cap \widetilde{\mathcal{V}}_{R+1} = \emptyset\,. \tag{2.24}$$

The typical rank of $R \times R \times 2$ arrays is equal to $\{R,\, R+1\}$, where both sets of rank values have positive volume. As in case 2, this implies that the sets $\mathcal{S}_R$ and $\widetilde{\mathcal{V}}_{R+1}$ have equal dimensionality. We are in the same situation as in case 8, with $\widetilde{\mathcal{W}}_R$ playing the role of $\mathcal{W}_R$, $\widetilde{\mathcal{V}}_{R+1}$ playing the role of $\mathcal{V}_{R+1}$, and $\underline{\mathbf{G}}$ playing the role of $\underline{\mathbf{H}}$. The remaining part of the explanation for cases 3, 5, and 9 is completely analogous to case 8 and is therefore omitted.

3. Diverging CP components for $p \times q \times 2$ arrays. In the previous section, we showed that if $\underline{\mathbf{X}}$ does not have a best rank-R approximation, this is due to the fact that the optimal solutions of problem (2.2) do not lie in $\mathcal{S}_R$, i.e., they have rank of at least $R+1$. From now on we assume there is one optimal solution $\widetilde{\underline{\mathbf{X}}}$ of problem (2.2). The general case is completely analogous. As a regularity condition, we assume that $\widetilde{\underline{\mathbf{X}}} \in \partial\mathcal{S}_R \cap \mathcal{S}_R^c$ lies in a subset $\mathcal{Q}_{R+1} \subset \partial\mathcal{S}_R \cap \mathcal{S}_R^c$ such that the dimensionality of $(\partial\mathcal{S}_R \cap \mathcal{S}_R^c) \backslash \mathcal{Q}_{R+1}$ is lower than the dimensionality of $\mathcal{Q}_{R+1}$ itself. In each of cases 2, 3, 5, 8, and 9 of Table 2.1, the set $\mathcal{Q}_{R+1}$ will be specified, and we show that if a sequence of CP updates $\underline{\mathbf{Y}}^{(n)}$ converges to $\widetilde{\underline{\mathbf{X}}}$, then $\underline{\mathbf{Y}}^{(n)}$ will exhibit diverging CP components as defined by (1.9) and (1.10). This implies that in these cases of Table 2.1, we conjecture that diverging CP components occur almost everywhere (case 2) or on a set of positive volume (cases 3, 5, 8, and 9). In Table 3.1 these conjectures are stated explicitly.

TABLE 3.1
Conjectures on the occurrence of diverging CP components when calculating a best rank-R approximation to generic $p \times q \times 2$ arrays. Here, $p \geq q \geq 2$ and $R \geq 2$.

Case	$\underline{\mathbf{X}} \in \mathbb{R}^{p\times q\times 2}$	$\text{rank}_\circ(\underline{\mathbf{X}})$	R	Diverging CP components?
2	$p = q$	$p+1$	$R = p$	almost everywhere
3	$p = q$	$p+1$	$R < p$	positive volume
5	$p = q$	p	$R < p$	positive volume
7	$p > q$	$\min(p, 2q)$	$\min(p, 2q) > R > q$	zero volume
8	$p > q$	$\min(p, 2q)$	$R = q$	positive volume
9	$p > q$	$\min(p, 2q)$	$R < q$	positive volume

Case 2. We have $p = q = R$ and $\text{rank}_\circ(\underline{\mathbf{X}}) = p+1$. We assume that $\widetilde{\underline{\mathbf{X}}}$ lies in $\mathcal{U}_R^{(12)}$, which is the set $\mathcal{Q}_{R+1}$ in this case. The set $\mathcal{U}_R^{(12)}$ is defined by (2.10). This implies that $\widetilde{\mathbf{X}}_2\widetilde{\mathbf{X}}_1^{-1}$ has p real eigenvalues and is not diagonalizable; see Stegeman [34]. From Proposition 1.2 it follows that $\text{rank}_\circ(\widetilde{\underline{\mathbf{X}}}) = p+k$, where k is the number of eigenvalues of $\widetilde{\mathbf{X}}_2\widetilde{\mathbf{X}}_1^{-1}$ with algebraic multiplicity larger than the geometric multiplicity. Suppose the sequence of CP updates $\underline{\mathbf{Y}}^{(n)}$ converges to $\widetilde{\underline{\mathbf{X}}}$. Since $\underline{\mathbf{Y}}^{(n)} \in \mathcal{S}_R$ and a singular $\mathbf{Y}_1^{(n)}$ does not occur in practice, it follows from Proposition 1.2 that

$\mathbf{Y}_2^{(n)}(\mathbf{Y}_1^{(n)})^{-1}$ has p real eigenvalues and is diagonalizable. Let $\mathbf{Y}_2^{(n)}(\mathbf{Y}_1^{(n)})^{-1}$ have the eigendecomposition

$$\mathbf{Y}_2^{(n)}(\mathbf{Y}_1^{(n)})^{-1} = \mathbf{K}^{(n)}\mathbf{\Lambda}^{(n)}(\mathbf{K}^{(n)})^{-1}\,, \tag{3.1}$$

where $\mathbf{K}^{(n)}$ has columns of length 1. A rank-p decomposition (1.6) of $\underline{\mathbf{Y}}^{(n)}$ is given by

$$\mathbf{A}^{(n)} = \mathbf{K}^{(n)}\,, \qquad (\mathbf{B}^{(n)})^T = \mathbf{\Omega}_b^{(n)}\,(\mathbf{K}^{(n)})^{-1}\mathbf{Y}_1^{(n)}\,, \tag{3.2}$$

$$\mathbf{C}^{(n)} = \begin{bmatrix} 1 & \cdots & 1 \\ \lambda_1^{(n)} & \cdots & \lambda_p^{(n)} \end{bmatrix} \mathbf{\Omega}_c^{(n)}\,, \qquad \mathbf{\Omega}^{(n)} = (\mathbf{\Omega}_b^{(n)}\,\mathbf{\Omega}_c^{(n)})^{-1}\,, \tag{3.3}$$

where $\mathbf{\Omega}_b^{(n)}$ and $\mathbf{\Omega}_c^{(n)}$ are the $p \times p$ diagonal matrices such that the columns of $\mathbf{B}^{(n)}$ and $\mathbf{C}^{(n)}$, respectively, have length 1. Hence, $\mathbf{C}_1^{(n)} = \mathbf{\Omega}_c^{(n)}$ and $\mathbf{C}_2^{(n)} = \mathbf{\Lambda}^{(n)}\mathbf{\Omega}_c^{(n)}$ in (1.6). It is clear that $k_{\mathbf{A}^{(n)}} = p$, $k_{\mathbf{B}^{(n)}} = p$, and $k_{\mathbf{C}^{(n)}} = 2$ if the eigenvalues in $\mathbf{\Lambda}^{(n)}$ are distinct. In this case, Kruskal's condition (1.7) holds and the rank-p decomposition (3.2)–(3.3) of $\underline{\mathbf{Y}}^{(n)}$ is unique. Since identical eigenvalues in $\mathbf{\Lambda}^{(n)}$ do not occur in practice, we assume they are all distinct (see Stegeman [34] for identical eigenvalues).

By continuity, the matrix $\mathbf{Y}_2^{(n)}(\mathbf{Y}_1^{(n)})^{-1}$ will converge to $\widetilde{\mathbf{X}}_2\widetilde{\mathbf{X}}_1^{-1}$. Since the latter matrix does not have p linearly independent eigenvectors and the rank-p decomposition (3.2)–(3.3) is unique, it follows that $\mathbf{A}^{(n)}$ in (3.2) converges to the singular matrix of eigenvectors of $\widetilde{\mathbf{X}}_2\widetilde{\mathbf{X}}_1^{-1}$. Let $I_1, \ldots, I_m \subset \{1, \ldots, R\}$ be the disjoint index sets of linearly dependent columns of the latter matrix such that each I_j contains the linearly dependent eigenvectors associated with a different eigenvalue of $\widetilde{\mathbf{X}}_2\widetilde{\mathbf{X}}_1^{-1}$ which has algebraic multiplicity larger than the geometric multiplicity. Then the columns $I_1, \ldots, I_m$ of the matrix $(\mathbf{K}^{(n)})^{-1}$ will become arbitrarily large as $\underline{\mathbf{Y}}^{(n)} \to \widetilde{\underline{\mathbf{X}}}$. The columns I_j of $\mathbf{C}^{(n)}$ will become identical, since these correspond to identical eigenvalues of $\widetilde{\mathbf{X}}_2\widetilde{\mathbf{X}}_1^{-1}$, $j = 1, \ldots, m$. It follows that in (3.2)–(3.3), we have $|\omega_r^{(n)}| \to \infty$ for all $r \in I_j$, $j = 1, \ldots, m$. Hence, (1.9) holds.

Next, we show that (1.10) also holds. For $j \in \{1, \ldots, m\}$, we consider the contribution of components I_j to the rank-p decomposition (3.2)–(3.3) for slices $\mathbf{Y}_1^{(n)}$ and $\mathbf{Y}_2^{(n)}$ separately. For $\mathbf{Y}_1^{(n)}$, the contribution of I_j equals

$$\mathbf{P}_j^{(n)} = \mathbf{K}_j^{(n)}\,(\mathbf{K}^{(n)})_j^{-1}\mathbf{Y}_1^{(n)}\,, \tag{3.4}$$

where $\mathbf{K}_j^{(n)}$ denotes columns I_j of $\mathbf{K}^{(n)}$ and $(\mathbf{K}^{(n)})_j^{-1}$ denotes rows I_j of $(\mathbf{K}^{(n)})^{-1}$. The limit point of $\mathbf{K}^{(n)}$ has columns I_j linearly dependent, while they are linearly independent of all its other columns. Hence, for n large enough, $\mathbf{K}^{(n)}$ will have columns I_j close to linear dependence but linearly independent of all its other columns. This, together with $\|\mathbf{Y}_1^{(n)}\|$ being bounded, yields that $\|\mathbf{P}_j^{(n)}\|$ in (3.4) is bounded.

For $\mathbf{Y}_2^{(n)}$, the contribution of I_j equals

$$\mathbf{Q}_j^{(n)} = \mathbf{K}_j^{(n)}\,\mathbf{\Lambda}_j^{(n)}\,(\mathbf{K}^{(n)})_j^{-1}\mathbf{Y}_1^{(n)}\,, \tag{3.5}$$

where $\mathbf{\Lambda}_j^{(n)}$ denotes the submatrix of $\mathbf{\Lambda}^{(n)}$ containing rows I_j and columns I_j. The diagonal matrix $\mathbf{\Lambda}_j^{(n)}$ converges to $\lambda\,\mathbf{I}_z$, where λ is the eigenvalue of $\widetilde{\mathbf{X}}_2\widetilde{\mathbf{X}}_1^{-1}$ associated

with eigenvectors I_j and $z = \text{card}(I_j)$. The matrix $\mathbf{K}_j^{(n)}$ contains the z eigenvectors associated with the eigenvalues of $\mathbf{Y}_2^{(n)}(\mathbf{Y}_1^{(n)})^{-1}$ on the diagonal of $\mathbf{\Lambda}_j^{(n)}$. As $\underline{\mathbf{Y}}^{(n)} \to \widetilde{\underline{\mathbf{X}}}$, the eigenvectors $\mathbf{K}_j^{(n)}$ converge to linear dependence but they remain linearly independent of all other eigenvectors. This, together with $\|\mathbf{Y}_1^{(n)}\|$ being bounded, yields that $\|\mathbf{Q}_j^{(n)}\|$ in (3.5) is bounded. Since

$$\left\| \sum_{r \in I_j} \omega_r^{(n)} \left(\mathbf{a}_r^{(n)} \circ \mathbf{b}_r^{(n)} \circ \mathbf{c}_r^{(n)} \right) \right\| = \sqrt{\|\mathbf{P}_j^{(n)}\|^2 + \|\mathbf{Q}_j^{(n)}\|^2}\,, \tag{3.6}$$

it follows that the left-hand side of (3.6) is bounded. Hence, (1.10) holds and the sequence of rank-p decompositions of $\underline{\mathbf{Y}}^{(n)}$ will exhibit diverging CP components as $\underline{\mathbf{Y}}^{(n)} \to \widetilde{\underline{\mathbf{X}}}$. Moreover, the groups $I_1, \ldots, I_m$ of diverging CP components and the number of components in each group are related to the eigenvalues and eigenvectors of $\widetilde{\mathbf{X}}_2\widetilde{\mathbf{X}}_1^{-1}$ as we have seen above.

Case 8. We have $p > q$ and $R = q$. We assume that the optimal solution $\widetilde{\underline{\mathbf{X}}}$ of problem (2.2) lies in $\mathcal{U}_R^{(12)}$, which is the set $\mathcal{Q}_{R+1}$ in this case. The set $\mathcal{U}_R^{(12)}$ is defined by (2.18). This implies that for any nonsingular $\mathbf{S}$ such that

$$\mathbf{S}\,\widetilde{\mathbf{X}}_1 = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{O} \end{bmatrix} \qquad \text{and} \qquad \mathbf{S}\,\widetilde{\mathbf{X}}_2 = \begin{bmatrix} \mathbf{H}_2 \\ \mathbf{O} \end{bmatrix}, \tag{3.7}$$

the $R \times R$ matrix $\mathbf{H}_2\mathbf{H}_1^{-1}$ has R real eigenvalues (which are not all distinct) and is not diagonalizable. By Proposition 1.2, $\text{rank}_\circ(\widetilde{\underline{\mathbf{X}}}) = \text{rank}_\circ(\underline{\mathbf{H}}) \geq R + 1$. Let $\underline{\mathbf{Y}}^{(n)}$ be a sequence of CP updates converging to $\widetilde{\underline{\mathbf{X}}}$. For a fixed $\mathbf{S}$ in (3.7), let $\mathbf{S}^{(n)}$ be such that it is nonsingular for all n, $\mathbf{S}^{(n)} \to \mathbf{S}$ and

$$\mathbf{S}^{(n)}\,\mathbf{Y}_1^{(n)} = \begin{bmatrix} \mathbf{H}_1^{(n)} \\ \mathbf{O} \end{bmatrix} \qquad \text{and} \qquad \mathbf{S}^{(n)}\,\mathbf{Y}_2^{(n)} = \begin{bmatrix} \mathbf{H}_2^{(n)} \\ \mathbf{O} \end{bmatrix}. \tag{3.8}$$

Then $\text{rank}_\circ(\underline{\mathbf{H}}^{(n)}) \leq R$ and $\underline{\mathbf{H}}^{(n)} \to \underline{\mathbf{H}}$ as $\underline{\mathbf{Y}}^{(n)} \to \widetilde{\underline{\mathbf{X}}}$. Since $\underline{\mathbf{H}}^{(n)}$, $\underline{\mathbf{H}} \in \mathbb{R}^{R \times R \times 2}$, it follows from case 2 above that the sequence $\underline{\mathbf{H}}^{(n)}$ will exhibit diverging CP components as it converges to $\underline{\mathbf{H}}$. Denote the unique rank-R decomposition of $\underline{\mathbf{H}}^{(n)}$ by $(\mathbf{A}^{(n)}, \mathbf{B}^{(n)}, \mathbf{C}^{(n)}, \mathbf{\Omega}^{(n)})$. Then the rank-$R$ decomposition of $\underline{\mathbf{Y}}^{(n)}$ is

$$\left((\mathbf{S}^{(n)})^{-1} \begin{bmatrix} \mathbf{A}^{(n)} \\ \mathbf{O} \end{bmatrix}, \mathbf{B}^{(n)}, \mathbf{C}^{(n)}, \mathbf{\Omega}^{(n)} \right). \tag{3.9}$$

The k-rank of $\mathbf{A}^{(n)}$ equals the k-rank of the first component matrix in (3.9). Hence, by virtue of Kruskal's condition (1.7) also the rank-R decomposition (3.9) is unique. Moreover, the decomposition will exhibit the same pattern of diverging CP components as $(\mathbf{A}^{(n)}, \mathbf{B}^{(n)}, \mathbf{C}^{(n)}, \mathbf{\Omega}^{(n)})$ when $\underline{\mathbf{Y}}^{(n)} \to \widetilde{\underline{\mathbf{X}}}$. Note that $\text{rank}_\circ(\underline{\mathbf{H}}) = \text{rank}_\circ(\widetilde{\underline{\mathbf{X}}})$ and Proposition 1.2 imply that the number of groups of diverging CP components does not depend on $\mathbf{S}$.

Cases 3, 5, and 9. We have $p \geq q$ and $R < q$. We assume that the optimal solution $\widetilde{\underline{\mathbf{X}}}$ of problem (2.2) satisfies

$$\mathbf{S}\,\widetilde{\mathbf{X}}_1\,\mathbf{T} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \qquad \text{and} \qquad \mathbf{S}\,\widetilde{\mathbf{X}}_2\,\mathbf{T} = \begin{bmatrix} \mathbf{G}_2 & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}, \tag{3.10}$$

where $\mathbf{S}$ and $\mathbf{T}$ are nonsingular and the $R \times R$ matrix $\mathbf{G}_2\mathbf{G}_1^{-1}$ has R real eigenvalues (which are not all distinct) and is not diagonalizable. By Proposition 1.2, $\text{rank}_\circ(\underline{\widetilde{\mathbf{X}}}) = \text{rank}_\circ(\underline{\mathbf{G}}) \geq R+1$. Let $\underline{\mathbf{Y}}^{(n)}$ be a sequence of CP updates converging to $\underline{\widetilde{\mathbf{X}}}$. For fixed $\mathbf{S}$ and $\mathbf{T}$ in (3.10), let $\mathbf{S}^{(n)}$ and $\mathbf{T}^{(n)}$ be such that they are nonsingular for all n, $\mathbf{S}^{(n)} \to \mathbf{S}$, $\mathbf{T}^{(n)} \to \mathbf{T}$, and

$$\text{(3.11)} \quad \mathbf{S}^{(n)}\,\mathbf{Y}_1^{(n)}\,\mathbf{T}^{(n)} = \begin{bmatrix} \mathbf{G}_1^{(n)} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \qquad \text{and} \qquad \mathbf{S}^{(n)}\,\mathbf{Y}_2^{(n)}\,\mathbf{T}^{(n)} = \begin{bmatrix} \mathbf{G}_2^{(n)} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}.$$

Then $\text{rank}_\circ(\underline{\mathbf{G}}^{(n)}) \leq R$ and $\underline{\mathbf{G}}^{(n)} \to \underline{\mathbf{G}}$ as $\underline{\mathbf{Y}}^{(n)} \to \underline{\widetilde{\mathbf{X}}}$. Since $\underline{\mathbf{G}}^{(n)}$, $\underline{\mathbf{G}} \in \mathbb{R}^{R\times R\times 2}$, it follows from case 2 that the sequence $\underline{\mathbf{G}}^{(n)}$ will exhibit diverging CP components as it converges to $\underline{\mathbf{G}}$. Denote the unique rank-R decomposition of $\underline{\mathbf{G}}^{(n)}$ by $(\mathbf{A}^{(n)}, \mathbf{B}^{(n)}, \mathbf{C}^{(n)}, \mathbf{\Omega}^{(n)})$. Then the rank-$R$ decomposition of $\underline{\mathbf{Y}}^{(n)}$ is

$$\text{(3.12)} \qquad \left((\mathbf{S}^{(n)})^{-1} \begin{bmatrix} \mathbf{A}^{(n)} \\ \mathbf{O} \end{bmatrix}, (\mathbf{T}^{(n)})^{-T} \begin{bmatrix} \mathbf{B}^{(n)} \\ \mathbf{O} \end{bmatrix}, \mathbf{C}^{(n)}, \mathbf{\Omega}^{(n)} \right).$$

The k-ranks of $\mathbf{A}^{(n)}$ and $\mathbf{B}^{(n)}$ equal the k-ranks of the first two component matrices in (3.12). Hence, by virtue of Kruskal's condition (1.7) the rank-R decomposition (3.12) is also unique. Moreover, the decomposition will exhibit the same pattern of diverging CP components as $(\mathbf{A}^{(n)}, \mathbf{B}^{(n)}, \mathbf{C}^{(n)}, \mathbf{\Omega}^{(n)})$ when $\underline{\mathbf{Y}}^{(n)} \to \underline{\widetilde{\mathbf{X}}}$. Note that $\text{rank}_\circ(\underline{\mathbf{G}}) = \text{rank}_\circ(\underline{\widetilde{\mathbf{X}}})$ and Proposition 1.2 imply that the number of groups of diverging CP components does not depend on $\mathbf{S}$ and $\mathbf{T}$.

4. Simulation results. Here, we illustrate the cases in Table 3.1 by trying to calculate (using a CP algorithm) a best rank-R approximation of random $p \times q \times 2$ arrays, the elements of which are sampled independently from the uniform distribution on $[-1, 1]$. We consider cases 3, 5, 8, and 9, in which we conjecture diverging CP components to occur on a set of positive volume, and case 7, in which we conjecture diverging CP components to occur on a set of zero volume. Simulation results in Stegeman [34] show that for case 2 diverging CP components always occur, which is in agreement with our conjecture in this case. Although different sampling distributions will give different results on the percentages of cases of diverging CP components, we feel that the outcomes presented below are useful to show that diverging CP components are a serious problem indeed. As a CP algorithm, we use the multilinear engine by Paatero [29].

The simulation results in Stegeman [34] have indicated that, for random $p \times p \times 2$ arrays $\underline{\mathbf{X}}$, problem (2.2) has a unique optimal solution $\underline{\widetilde{\mathbf{X}}}$. If $\underline{\widetilde{\mathbf{X}}} \in \mathcal{S}_R$, then problem (2.1) has a unique optimal solution and diverging CP components do not occur. If $\underline{\widetilde{\mathbf{X}}} \notin \mathcal{S}_R$, then it is approximated arbitrarily close by arrays in $\mathcal{S}_R$. Notice that if the component matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ have full k-rank (which they usually have), then Kruskal's condition (1.7) holds in cases 3, 5, 7, 8, and 9. Hence, if $\underline{\widetilde{\mathbf{X}}} \notin \mathcal{S}_R$, then the arrays in $\mathcal{S}_R$ close to $\underline{\widetilde{\mathbf{X}}}$ have a unique CP decomposition exhibiting diverging CP components.

The reasoning above implies that it suffices to use only one run of the CP algorithm for each array $\underline{\mathbf{X}}$ (with random starting values for the component matrices). For cases 3 and 5, we consider 100 random $4 \times 4 \times 2$ arrays $\underline{\mathbf{X}}$ and $R = 3$. The rank of $\underline{\mathbf{X}}$ depends on whether $\mathbf{X}_2\mathbf{X}_1^{-1}$ has 4 real eigenvalues (rank 4) or some complex eigenvalues (rank 5). In cases 8 and 9, we consider 100 random $5 \times 3 \times 2$ arrays and 50 random $5 \times 4 \times 2$ arrays, respectively, both with $R = 3$. For case 7, we consider

100 random $6 \times 3 \times 2$ arrays and $R = 5$. Table 4.1 gives the relative frequency of the occurrence of diverging CP components we encountered in these cases. As can be seen, diverging CP components occur quite often in cases 3, 5, 8, and 9, and they do not occur at all in case 7. Hence, the results in Table 4.1 are in agreement with our conjectures in Tabel 3.1 regarding the occurrence of diverging CP components.

TABLE 4.1
Results of calculating rank-R approximations to random $p \times q \times 2$ arrays in cases 3, 5, 7, 8, *and* 9 *of Table* 3.1.

Case	$\underline{\mathbf{X}}$: $p \times q \times 2$	rank($\underline{\mathbf{X}}$)	R	Diverging CP components
3	$4 \times 4 \times 2$	5	3	44 out of 70 (63%)
5	$4 \times 4 \times 2$	4	3	6 out of 30 (20%)
7	$6 \times 3 \times 2$	6	5	0 out of 100
8	$5 \times 3 \times 2$	5	3	51 out of 100 (51%)
9	$5 \times 4 \times 2$	5	3	24 out of 50 (48%)

5. Discussion. We have considered low-rank approximations to generic $p \times q \times 2$ arrays. For all combinations of p, q, and R, we presented conjectures on whether a best rank-R approximation exists almost everywhere, on a set of positive volume or on a set of zero volume. In the cases where no best rank-R approximation exists, this is due to the fact that the optimal boundary points of $\mathcal{S}_R$ (i.e., the optimal solutions of problem (2.2)) do not lie in $\mathcal{S}_R$ itself. We showed (under some regularity condition) that if a sequence of CP updates converges to such an optimal boundary point, it necessarily exhibits diverging CP components.

This explanation of diverging CP components confirms the statement of Kruskal, Harshman, and Lundy [24] that these occur due to the fact that the CP objective function does not attain its infimum, and that any sequence of CP updates of which the objective value is approaching the infimum must fail to converge and exhibits diverging CP components. Also, the concept of a sequence of CP updates converging to a boundary point $\widetilde{\underline{\mathbf{X}}} \notin \mathcal{S}_R$ can be found in Kruskal, Harshman, and Lundy [24] for the case $p = q = R = 2$. Whether diverging CP components always occur if there is no best rank-R approximation is still an open problem.

As in Stegeman [34], the occurrence of diverging CP components in the cases in Table 3.1 and their explanations are still valid when the Frobenius norm in the CP objective function is replaced by any other norm (e.g., weighted least squares or Gaussian maximum likelihood). This is because all norms on the finite-dimensional vector space are equivalent and induce the same (i.e., the Euclidian) topology.

Note that, as in Stegeman [34], the occurrence of diverging CP components in Table 3.1 does not depend on the algorithm used to minimize the CP objective function. Hence, modified CP algorithms designed to avoid diverging CP components are of no use here.

The diverging CP components in case 2 of Table 3.1 occur due to the two-valued typical rank of real-valued $p \times p \times 2$ arrays and the uniqueness of their rank-p decomposition; see the discussion in Stegeman [34]. Our results on diverging CP components for $p \times q \times 2$ arrays are based upon this. The typical rank of $p \times p \times 2$ arrays over the complex field is p. Therefore, the cases of diverging CP components described in this paper do not occur in the complex-valued CP model. However, also for the complex

field, a best low-rank approximation does not always exist. See the example in De Silva and Lim [10, Proposition 4.6], which carries over to the complex field.

In cases 3, 5, 8, and 9 of Table 3.1, diverging CP components occur due to the fact that we may transform the CP problem for $p \times q \times 2$ arrays to the lower-dimensional CP problem for $R \times R \times 2$ arrays, for which case 2 applies. This shows that a two-valued typical rank of the target array $\underline{\mathbf{X}}$ is not necessary for diverging CP components to occur. However, the two-valued typical rank for $R \times R \times 2$ arrays is still necessary for diverging CP components to occur.

Zijlstra and Kiers [43] observed that cases of two diverging CP components occur not only in CP but also in other variants of factor analysis. They show that two-way and three-way factor analysis models which yield diverging components necessarily have rotationally unique components. For the cases we have examined, diverging CP components always occur together with uniqueness of the CP solution. This raises the question whether (partial) uniqueness of the CP solution is necessary for diverging components to occur. Stegeman [35] has shown that this is not the case. Indeed, in the cases of $3 \times 3 \times 5$ arrays with symmetric slices and $8 \times 4 \times 3$ arrays, diverging CP components occur on a set of positive volume while the CP decompositions of the CP updates are not unique.

Acknowledgments. The author would like to thank the anonymous reviewers for valuable suggestions to improve the paper and for bringing to the author's attention the algebraic complexity literature dealing with array rank and border rank.

REFERENCES

[1] D. Bini, M. Capovani, F. Romani, and G. Lotti, *$O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication*, Inform. Process. Lett., 8 (1979), pp. 234–235.

[2] D. Bini, G. Lotti, and F. Romani, *Approximate solutions for the bilinear form computational problem*, SIAM J. Comput., 9 (1980), pp. 692–697.

[3] D. Bini, *Relations between exact and approximate bilinear algorithms. Applications*, Calcolo, 17 (1980), pp. 87–97.

[4] D. Bini, *Border rank of a $p \times q \times 2$ tensor and the optimal approximation of a pair of bilinear forms*, in Automata, Languages and Programming, J. W. de Bakker and J. van Leeuwen, eds., Lecture Notes in Comput. Sci. 85, Springer, New York, 1980, pp. 98–108.

[5] D. Bini, *Border rank of $m \times n \times (mn - q)$ tensors*, Linear Algebra Appl., 79 (1986), pp. 45–51.

[6] R. W. Brockett and D. Dobkin, *On the optimal evaluation of a set of bilinear forms*, Linear Algebra Appl., 19 (1978), pp. 207–235.

[7] P. Bürgisser, M. Clausen, and M. A. Shokrollahi, *Algebraic Complexity Theory*, Springer, Berlin, 1997.

[8] Y. Z. Cao, Z. P. Chen, C. Y. Mo, H. L. Wu, and R. Q. Yu, *A Parafac algorithm using penalty diagonalization error (PDE) for three-way data array resolution*, The Analyst, 125 (2000), pp. 2303–2310.

[9] J. D. Carroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart–Young decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[10] V. De Silva and L.-H. Lim, *Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem*, SCCM Technical report, 06-06, Stanford University, Palo Alto, CA, 2006; available online at http://www.sccm.stanford.edu/wrap/pubtech.html.

[11] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[12] F. R. Gantmacher, *The Theory of Matrices*, Volumes 1 and 2, Chelsea, New York, 1959.

[13] R. A. Harshman, *Foundations of the Parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[14] R. A. Harshman and M. E. Lundy, *Data preprocessing and the extended Parafac model*, in Research Methods for Multimode Data Analysis, H. G. Law, C. W. Snyder Jr., J. A. Hattie,

and R. P. McDonald, eds., Praeger, New York, 1984, pp. 216–284.
[15] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp. 164–189.
[16] F. L. Hitchcock, *Multiple invariants and generalized rank of a p-way matrix or tensor*, J. Math. Phys., 7 (1927), pp. 39–79.
[17] P. K. Hopke, P. Paatero, H. Jia, R. T. Ross, and R. A. Harshman, *Three-way (Parafac) factor analysis: Examination and comparison of alternative computational methods as applied to ill-conditioned data*, Chemom. Intel. Lab. Syst., 43 (1998), pp. 25–42.
[18] J. Ja'Ja', *Optimal evaluation of pairs of bilinear forms*, SIAM J. Comput., 8 (1979), pp. 443–462.
[19] J. Ja'Ja', *An addendum to Kronecker's theory of pencils*, SIAM J. Appl. Math., 37 (1979), pp. 700–712.
[20] L. Kronecker, *Algebraische reduction der schaaren bilinearer formen*, Sitzungber. Akad. Berlin (1890), pp. 763–776.
[21] P. M. Kroonenberg, *Three-Mode Principal Component Analysis*, DSWO, Leiden, The Netherlands, 1983.
[22] J. B. Kruskal, *Three-way arrays: Rank and uniqueness of trilinear decompositions, with applications to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.
[23] J. B. Kruskal, *Rank, decomposition, and uniqueness for* 3*-way and* N*-way arrays*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North-Holland, Amsterdam, 1989, pp. 7–18.
[24] J. B. Kruskal, R. A. Harshman, and M. E. Lundy, *How* 3*-MFA data can cause degenerate Parafac solutions, among other relationships*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North-Holland, Amsterdam, 1989, pp. 115–121 .
[25] J. M. Landsberg, *The border rank of the multiplication of* 2×2 *matrices in seven*, J. Amer. Math. Soc., 19 (2006), pp. 447–459.
[26] L.-H. Lim, *What's possible and what's impossible in tensor decompositions/approximation*, Talk at the Tensor Decomposition Workshop, Palo Alto, CA, 2004; available online at http://csmr.ca.sandia.gov/~tgkolda/tdw2004/.
[27] L.-H. Lim, *Optimal solutions to nonnegative Parafac/multilinear NMF always exist*, Talk at the Workshop on Tensor Decompositions and Applications, CIRM, Luminy, Marseille, France, 2005; available online at http://www.etis.ensea.fr/~wtda/.
[28] B. C. Mitchell and D. S. Burdick, *Slowly converging Parafac sequences: Swamps and two-factor degeneracies*, J. Chemometrics, 8 (1994), pp. 155–168.
[29] P. Paatero, *The Multilinear Engine—A table-driven least squares program for solving multilinear problems, including the n-way Parallel Factor Analysis model*, J. Comput. Graph. Statist., 8 (1999), pp. 854–888.
[30] P. Paatero, *Construction and analysis of degenerate Parafac models*, J. Chemometrics, 14 (2000), pp. 285–299.
[31] W. S. Rayens and B. C. Mitchell, *Two-factor degeneracies and a stabilization of Parafac*, Chemometrics Intel. Lab. Syst., 38 (1997), pp. 173–181.
[32] N. D. Sidiropoulos, *Low-rank decomposition of multi-way arrays: A signal processing perspective*, Invited Plenary Lecture at the IEEE Sensor Array and Multichannel (SAM) Signal Processing Workshop, Sitges, Barcelona, Spain, 2004. Available online at http://www.telecom.tuc.gr/~nikos/.
[33] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis: Applications in the Chemical Sciences*, John Wiley & Sons, New York, 2004.
[34] A. Stegeman, *Degeneracy in Candecomp/Parafac explained for* $p \times p \times 2$ *arrays of rank* $p+1$ *or higher*, Psychometrika, 71 (2006), pp. 483–501.
[35] A. Stegeman, *Degeneracy in Candecomp/Parafac explained for several three-sliced arrays with a two-valued typical rank*, Psychometrika, (2008), in press.
[36] V. Strassen, *Vermeidung von divisionen*, J. Reine Angew. Math., 246 (1973), pp. 184–202.
[37] J. M. F ten Berge, H. A. L. Kiers, and J. De Leeuw, *Explicit Candecomp/Parafac solutions for a contrived* $2 \times 2 \times 2$ *array of rank three*, Psychometrika, 53 (1988), pp. 579–584.
[38] J. M. F. ten Berge, *Kruskal's polynomial for* $2 \times 2 \times 2$ *arrays and a generalization to* $2 \times n \times n$ *arrays*, Psychometrika, 56 (1991), pp. 631–636.
[39] J. M. F. ten Berge and H. A. L. Kiers, *Simplicity of core arrays in three-way principal component analysis and the typical rank of* $p \times q \times 2$ *arrays*, Linear Algebra Appl., 294 (1999), pp. 169–179.
[40] J. M. F. ten Berge, *Partial uniqueness in Candecomp/Parafac*, J. Chemometrics, 18 (2004), pp. 12–16.

[41] G. Tomasi and R. Bro, *A Comparison of algorithms for fitting the Parafac model*, Comput. Statist. Data Anal., 50 (2006), pp. 1700–1734.
[42] K. Weierstrass, *Zur theorie der bilinearen und quadratischen formen*, Monatsber. Akad. Wiss. Berlin, (1868), pp. 310–338.
[43] B. J. H. Zijlstra and H. A. L. Kiers, *Degenerate solutions obtained from several variants of factor analysis*, J. Chemometrics, 16 (2002), pp. 596–605.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1008–1021

FAST MULTILINEAR SINGULAR VALUE DECOMPOSITION FOR STRUCTURED TENSORS*

ROLAND BADEAU† AND RÉMY BOYER‡

Abstract. The higher-order singular value decomposition (HOSVD) is a generalization of the singular value decomposition (SVD) to higher-order tensors (i.e., arrays with more than two indices) and plays an important role in various domains. Unfortunately, this decomposition is computationally demanding. Indeed, the HOSVD of a third-order tensor involves the computation of the SVD of three matrices, which are referred to as "modes" or "matrix unfoldings." In this paper, we present fast algorithms for computing the full and the rank-truncated HOSVD of third-order structured (symmetric, Toeplitz, and Hankel) tensors. These algorithms are derived by considering two specific ways to unfold a structured tensor, leading to structured matrix unfoldings whose SVD can be efficiently computed.

Key words. multilinear SVD, fast algorithms, structured and unstructured tensors

AMS subject classifications. 15A69, 15A18, 15A21

DOI. 10.1137/060655936

1. Introduction.

The subject of multilinear decomposition is now mature [5, 19]. There are essentially two families. The first one is known under the name of CANDECOMP/PARAFAC (CANonical DECOMPosition or PARAllel FACtors model) and was independently proposed in [4, 8]. This decomposition is very useful in several applications and is linked to the tensor rank [9]. The second one is related to the multidimensional rank [6] and is known under the name of Tucker decomposition [21]. This decomposition is a more general form which is often used. Orthogonality constraints are not required in the general Tucker decomposition but if needed, one can refer to the higher-order singular value decomposition (HOSVD) [6] or multilinear SVD.

The HOSVD is a generalization of the SVD to higher-order tensors (i.e., arrays with more than two indices). This decomposition plays an important role in various domains, such as harmonic retrieval [17], image processing [10], telecommunications, biomedical applications (magnetic resonance imaging and electrocardiography), web search [20], computer facial recognition [23], handwriting analysis [18], and statistical methods involving independent component analysis (ICA) [6].

In [14], it was shown that the HOSVD of a third-order tensor involves the computation of the SVD of three matrices called modes, leading to a high computational cost. A first approach for reducing the complexity of tensor-based methods consists of a dimensionality reduction: only the principal components of the HOSVD are calculated, leading to the *rank-truncated* HOSVD. In this paper, we present a standard and fast algorithm for calculating the full and rank-truncated HOSVD, which computes only the left factors of the three SVD's. Next, we focus on structured tensors, such as

*Received by the editors March 31, 2006; accepted for publication (in revised form) by N. Mastronardi April 11, 2007; published electronically September 25, 2008. This work has been partially presented at the IEEE ICASSP 2006 conference [3].

http://www.siam.org/journals/simax/30-3/65593.html

†GET - Télécom Paris (ENST), Département TSI, 46 rue Barrault, 75634 Paris Cedex 13, France (roland.badeau@enst.fr).

‡Laboratoire des Signaux et Systémes (LSS), CNRS, Université Paris XI (UPS), SUPELEC, Gif-Sur-Yvette, France (remy.boyer@lss.supelec.fr).

symmetric and Toeplitz tensors, which naturally arise in signal processing methods involving higher-order statistics [11, Chapter 9], and Hankel tensors [17], introduced in the context of the harmonic retrieval problem [15], which is at the heart of many signal processing applications. To the best of our knowledge, there are no specific HOSVD algorithms proposed in the literature for exploiting tensors structures. In this paper, however, we show that such tensors can be efficiently decomposed. We first observe that standard unfoldings [14, 12] do not present a particularly noticeable structure even in the case of structured tensors. Consequently, we introduce two different ways to unfold a structured tensor which clarify the link between structured modes and structured tensors. By doing this, we can exploit fast product techniques [7]. A second point of this work concerns Hankel and symmetric tensors. The modes of these structured tensors are column-redundant so it is possible to reduce the computational cost of the HOSVD algorithm by taking the redundant structure of each mode into account. Finally, our fastest implementation of the rank-truncated HOSVD (dedicated to Hankel tensors) has a quasilinear complexity with respect to the tensor dimension.

Note that, for applications involving very large tensor dimensions, an even lower complexity may be required. In this case, one may be interested in rank-revealing tensor decompositions which can be computed faster than the rank-truncated HOSVD. Such an approach is developed in [16], based on cross approximation techniques which are derived from LU factorizations [22, 2]. An algorithm is proposed which provides a Tucker-like low rank approximation of unstructured cube tensors, the complexity of which is linear with respect to the tensor dimension in many cases [16]. This linear complexity is nevertheless obtained via an approximated rank reduction, in comparison with an exact Tucker decomposition such as the HOSVD.

2. Preliminaries in multilinear algebra. We present some basic definitions in the context of third-order tensor algebra. These definitions can be extended to order greater than three, and we refer the interested reader to [5, 6] for instance.

Tucker's product. The Tucker's product, also called the s-mode product, of a third-order complex-valued tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ by a matrix $B \in \mathbb{C}^{J_s \times I_s}$ for $s \in [1:3]$ is defined according to the following:

$$[\mathcal{A} \times_1 B]_{j_1 i_2 i_3} = \sum_{i_1=0}^{I_1-1} [\mathcal{A}]_{i_1 i_2 i_3} [B]_{j_1 i_1}, \tag{2.1}$$

$$[\mathcal{A} \times_2 B]_{i_1 j_2 i_3} = \sum_{i_2=0}^{I_2-1} [\mathcal{A}]_{i_1 i_2 i_3} [B]_{j_2 i_2}, \tag{2.2}$$

$$[\mathcal{A} \times_3 B]_{i_1 i_2 j_3} = \sum_{i_3=0}^{I_3-1} [\mathcal{A}]_{i_1 i_2 i_3} [B]_{j_3 i_3}, \tag{2.3}$$

where we denoted the entries of $\mathcal{A}$ by $[\mathcal{A}]_{i_1 i_2 i_3}$ with $i_s \in \{0 \dots I_s - 1\}$. We have the following properties:

$$\mathcal{A} \times_s B \times_{s'} C = (\mathcal{A} \times_s B) \times_{s'} C = (\mathcal{A} \times_{s'} C) \times_s B, \tag{2.4}$$

$$(\mathcal{A} \times_s B) \times_s C = \mathcal{A} \times_s (BC). \tag{2.5}$$

Mode of a tensor. There are several ways to represent an $I_1 \times I_2 \times I_3$ third-order complex-valued tensor $\mathcal{A}$ as a collection of matrices.

DEFINITION 2.1. *The modes (also called "matrix unfoldings") A_1, A_2, A_3 are usually defined as follows:*

$$[A_1]_{i_1, i_2 I_3 + i_3} = [\mathcal{A}]_{i_1 i_2 i_3}, \tag{2.6}$$

$$[A_2]_{i_2, i_3 I_1 + i_1} = [\mathcal{A}]_{i_1 i_2 i_3}, \tag{2.7}$$

$$[A_3]_{i_3, i_1 I_2 + i_2} = [\mathcal{A}]_{i_1 i_2 i_3}. \tag{2.8}$$

These matrices are of dimension $(I_1 \times I_2 I_3)$, $(I_2 \times I_3 I_1)$, $(I_3 \times I_1 I_2)$, respectively.

The dimensions of the vector spaces generated by the columns of the modes of $\mathcal{A}$ are called column rank (or 1-mode rank) R_1, row rank (or 2-mode rank) R_2, and 3-mode rank R_3, respectively.

2.1. Multilinear SVD (HOSVD).

THEOREM 2.2 (third-order SVD [6, 21]). *Every $I_1 \times I_2 \times I_3$ tensor $\mathcal{A}$ can be written as the product*

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}, \tag{2.9}$$

where $\times_s$ represents the Tucker s-mode product [6]*, $U^{(s)}$ is a unitary $I_s \times I_s$ matrix, and $\mathcal{S}$ is an all-orthogonal and ordered $I_1 \times I_2 \times I_3$ tensor. All-orthogonality means that the matrices $S_{i_s=\alpha}$, obtained by fixing the sth index to α, are mutually orthogonal with respect to (w.r.t.) the standard inner product. Ordering means that $\|S_{i_s=0}\| \geqslant \|S_{i_s=1}\| \geqslant \cdots \geqslant \|S_{i_s=I_s-1}\|$ for all possible values of s. The Frobenius-norms $\|S_{i_s=i}\|$, symbolized by $\sigma_i^{(s)}$, are the s-mode singular values of $\mathcal{A}$ and the columns of $U^{(s)}$ are the s-mode singular factors.*

This decomposition is a generalization of the SVD because the diagonality of the matrix containing the singular values (in the matrix case) is a special case of all-orthogonality. Also, the HOSVD of a second-order tensor (matrix) yields the matrix SVD, up to trivial indeterminacies. The matrix of s-mode singular factors, $U^{(s)}$, can be found as the matrix of left singular vectors of the mode A_s, defined in (2.6)–(2.8). The s-mode singular values correspond to the singular values of this matrix unfolding. Note that the s-mode singular factors of a tensor, corresponding to the nonzero s-mode singular values, form an orthonormal basis for its s-mode vector subspace, as in the matrix case.

The core tensor $\mathcal{S}$ can then be computed (if needed) by bringing the matrices of s-mode singular factors to the left side of (2.9):

$$\mathcal{S} = \mathcal{A} \times_1 {U^{(1)}}^H \times_2 {U^{(2)}}^H \times_3 {U^{(3)}}^H, \tag{2.10}$$

where $(.)^H$ denotes the conjugate transpose.

Mode decompositions. Expression (2.9) can be written in terms of modes as follows:

$$A_1 = U^{(1)} S_1 \left(U^{(3)} \otimes U^{(2)}\right)^H,$$

$$A_2 = U^{(2)} S_2 \left(U^{(3)} \otimes U^{(1)}\right)^H,$$

$$A_3 = U^{(3)} S_3 \left(U^{(1)} \otimes U^{(2)}\right)^H,$$

where $\otimes$ denotes the Kronecker product, and S_1, S_2, and S_3 denote, respectively, the first, second, and third modes of the core tensor $\mathcal{S}$.

2.2. HOSVD algorithm for unstructured tensors. In this section we present an efficient implementation of the HOSVD in the general framework of unstructured tensors, from which our fast algorithms for structured tensors will be derived in section 4. Let $I = \frac{1}{3}(I_1 + I_2 + I_3)$. The computational costs of the various algorithms presented below are related to the flop (*floating point operation*) count. For example, a dot product of I-dimensional vectors involves $2I$ flops (I multiplications plus I additions).

The calculation of the HOSVD of tensor $\mathcal{A}$ requires the computation, for all $s \in [1:3]$, of the left factor $U^{(s)}$ in the full SVD of matrix A_s, as defined previously. Note that, in many applications we are interested in computing the HOSVD truncated at ranks (M_1, M_2, M_3), which means that we compute only the M_s first columns of the matrix $U^{(s)}$ (M_s is often supposed to be much lower than I_s). We will suppose throughout this paper that this possibly truncated SVD is computed by means of the orthogonal iteration method, although other algorithms such as the Golub–Reinsch SVD and R-SVD [7, pp. 253–254] could also be applied. When computing only the $n \times r$ left factor U in the rank r-truncated SVD of an $n \times m$ matrix A with $n < m$, the orthogonal iteration method consists of recursively computing the $n \times r$ matrix $B_i = A(A^H U_{i-1})$, involving $2r$ matrix/vector products, and the QR factorization $B_i = U_i R_i$ of this $n \times r$ matrix [7, pp. 410–411]. Thus the computational cost of one iteration is $2r\, c(n,m) + 2r^2 n$ flops, where $c(n,m) = 2nm$ is the cost of 1 matrix/vector product, and $2r^2 n$ is the cost of 1 QR factorization [7, pp. 231–232]. Besides, the s-mode A_s has $n = I_s$ rows and $m = \prod_{s' \neq s} I_{s'}$ columns. Assuming that $\prod_{s' \neq s} I_{s'}$ is much greater than I_s, the dominant cost of one iteration for computing $U^{(s)}$ is $4M_s I_1 I_2 I_3$ flops. Finally, the Tucker product (2.10) can be computed by folding, for instance, its first mode given by

$$S_1 = {U^{(1)}}^H A_1 \left(U^{(3)} \otimes U^{(2)} \right). \tag{2.11}$$

A fast implementation of (2.11) was proposed in [1], whose complexity is $6M_s I_1 I_2 I_3$ flops, where $M = \frac{1}{3}(M_1 + M_2 + M_3)$. Note that the computation of the Tucker product is generally not needed in applications, which is why it will be omitted in the following developments.

The computational cost of the full and rank-truncated HOSVD is summarized in Table 2.1 (the full HOSVD is the same as the rank-truncated HOSVD with $M_s = I_s$ for all $s = 1, 2, 3$). In this table and afterward, the global cost is provided as a maximum w.r.t. I_1, I_2, I_3, under the constraint $I_1 + I_2 + I_3 = 3I$. In particular, the maximal complexity per iteration is obtained for cube tensors ($I_1 = I_2 = I_3 = I$) and equals $12MI^3$.

TABLE 2.1
HOSVD Algorithm for unstructured tensors.

(the cost corresponds to a single iteration of the orthogonal iteration method)

Operation	Cost per iteration
SVD of A_1	$4M_1 I_1 I_2 I_3$
SVD of A_2	$4M_2 I_1 I_2 I_3$
SVD of A_3	$4M_3 I_1 I_2 I_3$
Global cost	$12MI^3$

3. Structured tensors and reordered tensor modes. In this section, we present three tensor structures which are usual in many applications. Next, we intro-

duce new reordered tensor modes which clarify the link between structured tensors and structured modes.

3.1. Structured tensors.

DEFINITION 3.1 (Toeplitz tensors). *A* Toeplitz *tensor is a structured tensor which satisfies the following property: for all* $i_1 \in \{0 \dots I_1 - 1\}$, $i_2 \in \{0 \dots I_2 - 1\}$, $i_3 \in \{0 \dots I_3 - 1\}$ *for all* $k \in \{0 \dots \min(I_1 - i_1, I_2 - i_2, I_3 - i_3) - 1\}$,

$$[\mathcal{A}]_{i_1+k,i_2+k,i_3+k} = [\mathcal{A}]_{i_1 i_2 i_3}.$$

Below, any permutation of 3 elements will be denoted $\pi = (\pi_1, \pi_2, \pi_3)$, where $\pi_1, \pi_2, \pi_3 \in \{1, 2, 3\}$, according to the following definition:

$$\pi : \quad (i_1, i_2, i_3) \quad \mapsto \quad (i_{\pi_1}, i_{\pi_2}, i_{\pi_3}).$$

DEFINITION 3.2 (symmetric tensors). *A cube* $(I \times I \times I)$ *tensor* $\mathcal{A}$*, which is unchanged by any permutation* π*, is called a* symmetric *tensor:*

$$\forall i_1, i_2, i_3 \in \{0, \dots, I-1\}, \ [\mathcal{A}]_{\pi(i_1,i_2,i_3)} = [\mathcal{A}]_{i_1 i_2 i_3}.$$

Example 1 (fast higher-order PCA for real moment and cumulant). The HOSVD can be viewed (cf. [13]) as a higher-order principal component analysis (PCA). This technique is often used as a data dimensional reduction for moment and cumulant tensors [6]. Third-order moment and cumulant tensors are defined according to

$$[\mathcal{M}]_{t_1 t_2 t_3} = E\{x(t_1)x(t_2)x(t_3)\}, \tag{3.1}$$

$$\begin{aligned} [\mathcal{C}]_{t_1 t_2 t_3} = \ & E\{x(t_1)x(t_2)x(t_3)\} + 2E\{x(t_1)\}E\{x(t_2)\}E\{x(t_3)\} \\ - \ & E\{x(t_1)\}E\{x(t_2)x(t_3)\} - E\{x(t_2)\}E\{x(t_1)x(t_3)\} \\ - \ & E\{x(t_3)\}E\{x(t_1)x(t_2)\}, \end{aligned} \tag{3.2}$$

where $t_1, t_2, t_3 \in \{0, \dots, I-1\}$, and $x(t)$ is a real random process.

Moment and cumulant tensors, defined in (3.1) and (3.2), are symmetric tensors according to Definition 3.2. The proof is straightforward and can be generalized to larger orders [5]. Moreover, if $x(t)$ is a third-order stationary process, the moment and cumulant tensors defined in (3.1) and (3.2) are third-order Toeplitz tensors according to Definition 3.2. Indeed, if $x(t)$ is a stationary process, its probability distribution is invariant to temporal translations. This property implies $[\mathcal{C}]_{t+i_1,t+i_2,t+i_3} = [\mathcal{C}]_{i_1 i_2 i_3}$.

DEFINITION 3.3 (Hankel tensors). *A Hankel tensor is a structured tensor whose coefficients* $[\mathcal{A}]_{i_1 i_2 i_3}$ *depend only on* $i_1 + i_2 + i_3$.

Note that a cube Hankel tensor is symmetric. Hankel tensors were introduced in [17] in the context of the harmonic retrieval problem [15]. This problem is at the heart of many signal processing applications.

Example 2 (definition and properties of the harmonic model). We consider the complex harmonic model defined according to

$$x_n = \sum_{m=1}^{M} \alpha_m z_m^n, \quad \text{for} \quad n \in [0 : N-1], \tag{3.3}$$

where N is the analysis duration and M is the known number of components, $z_m = e^{\delta_m + i\phi_m}$ is called the mth pole of x_n where $i = \sqrt{-1}$, ϕ_m is called the mth angular-frequency belonging to $(-\pi, \pi]$, and δ_m is the mth damping factor. In what follows,

we assume that all the poles are distinct. In addition, $\alpha_m = a_m e^{i\phi_m}$ is the nonzero mth complex amplitude, i.e., $a_m \neq 0 \, for \, all \, m$. Besides, we define the Vandermonde matrices $Z^{(I_1)}$, $Z^{(I_2)}$, and $Z^{(I_3)}$ associated to model (3.3), according to $[Z^{(I_s)}]_{n,m} = z_m^n$, and we assume that $M \leq \min(I_1, I_2, I_3)$. Then the Hankel tensor $[\mathcal{A}]_{i_1 i_2 i_3} = x_{(i_1+i_2+i_3)}$ associated to model (3.3) is diagonalizable according to

$$\mathcal{A} = \mathcal{D} \times_1 Z^{(I_1)} \times_2 Z^{(I_2)} \times_3 Z^{(I_3)},$$

where $\times_i$ denotes the ith Tucker's product and

$$[\mathcal{D}]_{jk\ell} = \begin{cases} \alpha_j & \text{if } j = k = \ell, \\ 0 & \text{otherwise} \end{cases}$$

is a hyper-cubic $M \times M \times M$ super-diagonal core tensor. As a consequence, the Hankel tensor $\mathcal{A}$ is a rank-(M, M, M) tensor. Following standard subspace-based parametric estimation methods, the harmonic model can then be estimated by computing the rank M-truncated HOSVD of tensor $\mathcal{A}$ [17].

3.2. Modes of structured tensors. As mentioned in the introduction, standard unfoldings of structured tensors [14, 12] do not present a particularly noticeable structure. Consequently, we introduce in this section two different ways to unfold a structured tensor which clarify the link between structured modes and structured tensors.

Example 3. Consider the $4 \times 4 \times 4$ symmetric and Toeplitz tensor $[\mathcal{A}]_{ijk} = 3 * \max(i, j, k) - \text{sum}(i, j, k)$. The classical 1-mode is formed of four symmetric submatrices:

$$A_1 = \left[\begin{bmatrix} 0 & 2 & 4 & 6 \\ 2 & 1 & 3 & 5 \\ 4 & 3 & 2 & 4 \\ 6 & 5 & 4 & 3 \end{bmatrix} \begin{bmatrix} 2 & 1 & 3 & 5 \\ 1 & 0 & 2 & 4 \\ 3 & 2 & 1 & 3 \\ 5 & 4 & 3 & 2 \end{bmatrix} \begin{bmatrix} 4 & 3 & 2 & 4 \\ 3 & 2 & 1 & 3 \\ 2 & 1 & 0 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 6 & 5 & 4 & 3 \\ 5 & 4 & 3 & 2 \\ 4 & 3 & 2 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \right].$$

However, by permuting its columns, we define another mode A_1' (referred to below as the *type*-1 *reordered tensor mode*), which is formed of four Toeplitz matrices T_0, T_1, T_2, and T_3 (referred to below as the *type*-1 *oblique submatrices*):

$$A_1' = \left[\underbrace{\begin{bmatrix} 6 \\ 5 \\ 4 \\ 3 \end{bmatrix}}_{T_3} \underbrace{\begin{bmatrix} 4 & 5 \\ 3 & 4 \\ 2 & 3 \\ 4 & 2 \end{bmatrix}}_{T_2} \underbrace{\begin{bmatrix} 2 & 3 & 4 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \\ 5 & 3 & 1 \end{bmatrix}}_{T_1} \underbrace{\begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 2 \\ 4 & 2 & 0 & 1 \\ 6 & 4 & 2 & 0 \end{bmatrix}}_{T_0} \underbrace{\begin{bmatrix} 2 & 3 & 4 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \\ 5 & 3 & 1 \end{bmatrix}}_{T_1} \underbrace{\begin{bmatrix} 4 & 5 \\ 3 & 4 \\ 2 & 3 \\ 4 & 2 \end{bmatrix}}_{T_2} \underbrace{\begin{bmatrix} 6 \\ 5 \\ 4 \\ 3 \end{bmatrix}}_{T_3} \right].$$

It can be noted that this reordered mode satisfies an axial blockwise symmetry with respect to its central oblique submatrix T_0. Obviously, the left singular factor $U^{(1)}$ in the SVD of A_1 is the same as the left singular factor in the SVD of A_1', since both matrices have the same columns. However, we will show that the SVD of A_1' can be computed efficiently, by exploiting the Toeplitz structure of the oblique submatrices T_k.

Example 4. Consider the $4 \times 4 \times 4$ Hankel tensor $[\mathcal{A}]_{ijk} = i + j + k$. The standard 1-mode is formed of four Hankel submatrices:

$$A_1 = \left[\begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \\ 5 & 6 & 7 & 8 \end{bmatrix} \begin{bmatrix} 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \\ 5 & 6 & 7 & 8 \\ 6 & 7 & 8 & 9 \end{bmatrix} \right].$$

However, by permuting its columns, we define another mode A_1'' (referred to below as the *type-2 reordered tensor mode*), which is formed of 7 rank-1 matrices $R_0,\dots,R_6$ (referred to below as the *type-2 oblique submatrices*):

$$A_1'' = \left[\underbrace{\begin{bmatrix}0\\1\\2\\3\end{bmatrix}}_{R_0} \underbrace{\begin{bmatrix}1&1\\2&2\\3&3\\4&4\end{bmatrix}}_{R_1} \underbrace{\begin{bmatrix}2&2&2\\3&3&3\\4&4&4\\5&5&5\end{bmatrix}}_{R_2} \underbrace{\begin{bmatrix}3&3&3&3\\4&4&4&4\\5&5&5&5\\6&6&6&6\end{bmatrix}}_{R_3} \underbrace{\begin{bmatrix}4&4&4\\5&5&5\\6&6&6\\7&7&7\end{bmatrix}}_{R_4} \underbrace{\begin{bmatrix}5&5\\6&6\\7&7\\8&8\end{bmatrix}}_{R_5} \underbrace{\begin{bmatrix}6\\7\\8\\9\end{bmatrix}}_{R_6} \right].$$

Again, the left singular factor in the SVD of A_1 is the same as the left singular factor in the SVD of A_1'', since both matrices have the same columns. However, we will show that the SVD of A_1'' can be computed efficiently by exploiting the rank-1 structure of the oblique submatrices R_k and the Hankel structure of the matrix obtained by removing the repeated columns in A_1''.

In the following section, the type-1 and type-2 oblique submatrices will be defined in the general case by "slicing" a third-order tensor according to two different oblique directions, as shown in Figure 3.1.

3.2.1. Oblique submatrices of a tensor.

DEFINITION 3.4 (type-1 and type-2 oblique submatrices of a tensor). *For any permutation π, the oblique submatrices of a tensor $\mathcal{A}$ are defined as follows:*

- *For all $k \in \{0,\dots,I_{\pi_3}-1\}$, let $J^{(1)}_{(\pi_2,\pi_3)}(k) = \min(I_{\pi_2}, I_{\pi_3}-k)$. The coefficients of the kth type-1 oblique $I_{\pi_1} \times J^{(1)}_{(\pi_2,\pi_3)}(k)$ submatrix of $\mathcal{A}$ are*

$$[T_k^{(\pi)}]_{ij} = [\mathcal{A}]_{\pi^{-1}(i,j,k+j)}, \tag{3.4}$$

  where $0 \le i \le I_{\pi_1}-1$ and $0 \le j \le J^{(1)}_{(\pi_2,\pi_3)}(k)-1$.
- *For all $k \in \{0,\dots,I_{\pi_2}+I_{\pi_3}-2\}$, let*

$$J^{(2)}_{(\pi_2,\pi_3)}(k) = \min(I_{\pi_2}, I_{\pi_3}, 1+k, I_{\pi_2}+I_{\pi_3}-1-k). \tag{3.5}$$

  The coefficients of the $I_{\pi_1} \times J^{(2)}_{(\pi_2,\pi_3)}(k)$ type-2 oblique submatrix of $\mathcal{A}$ are

$$[R_k^{(\pi)}]_{ij} = [\mathcal{A}]_{\pi^{-1}(i,\ \max(k-I_{\pi_3}+1,0)+j,\ \min(k,I_{\pi_3}-1)-j)}, \tag{3.6}$$

  where $0 \le i \le I_{\pi_1}-1$ and $0 \le j \le J^{(2)}_{(\pi_2,\pi_3)}(k)-1$.

PROPOSITION 3.5.

1. *If $\mathcal{A}$ is an $(I \times I \times I)$ symmetric tensor, then for all $k \in \{0,\dots,I-1\}$, all type-1 oblique submatrices $T_k^{(\pi)}$ are equal (*i.e.*, for all k, $T_k^{(\pi)} = T_k$ does not depend on π).*
2. *If $\mathcal{A}$ is a Toeplitz tensor, then for all permutation π and index $k \in \{0,\dots,I_{\pi_3}-1\}$, the type-1 oblique submatrix $T_k^{(\pi)}$ is Toeplitz.*
3. *If $\mathcal{A}$ is a Hankel tensor, all columns of the type-2 oblique submatrix $R_k^{(\pi)}$ are equal.*

Proof.

1. If the tensor $\mathcal{A}$ is symmetric, then (3.4) yields $[T_k^{(\pi)}]_{ij} = [\mathcal{A}]_{i,j,k+j}$, which does not depend on π.

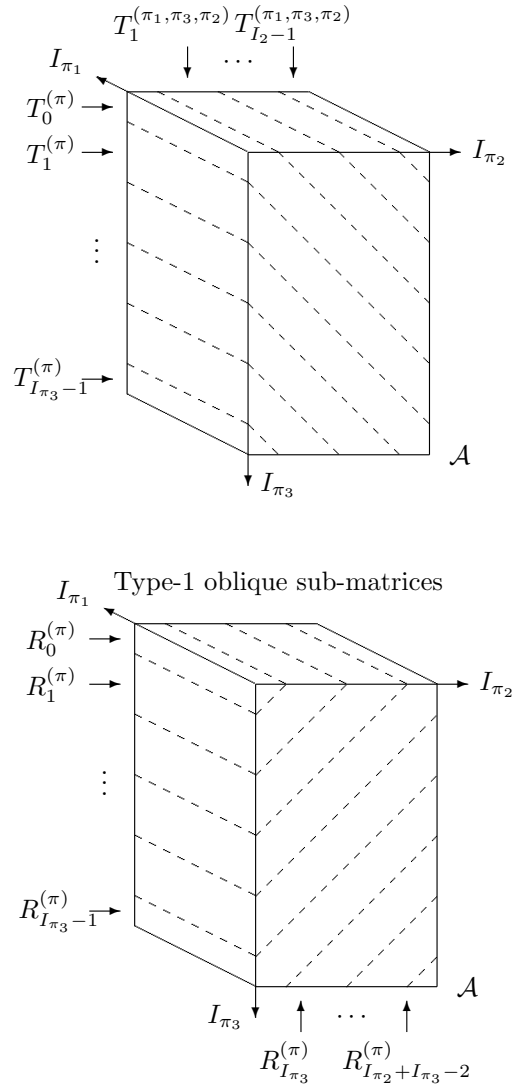


FIG. 3.1. *Type-*1 *and type-*2 *oblique submatrices of a tensor.*

2. Applying (3.4) to $i+1$ and $j+1$ (for all $0 \le i < I_{\pi_1} - 1$ and $0 \le j < J^{(1)}_{(\pi_2,\pi_3)}(k) - 1$) yields $[T_k^{(\pi)}]_{i+1,j+1} = [\mathcal{A}]_{\pi^{-1}(i+1,j+1,k+j+1)}$. However, since the tensor $\mathcal{A}$ is Toeplitz, $[\mathcal{A}]_{\pi^{-1}(i+1,j+1,k+j+1)} = [\mathcal{A}]_{\pi^{-1}(i,j,k+j)} = [T_k^{(\pi)}]_{ij}$. Therefore $[T_k^{(\pi)}]_{i+1,j+1} = [T_k^{(\pi)}]_{ij}$, which means that the matrix T_k is Toeplitz.
3. If $[\mathcal{A}]_{i_1 i_2 i_3}$ is of the form $[\mathcal{A}]_{i_1 i_2 i_3} = x_{(i_1+i_2+i_3)}$, then (3.6) shows that for all permutation π and index $k \in \{0, \dots, I_{\pi_2} + I_{\pi_3} - 2\}$, $[R_k^{(\pi)}]_{ij} = x_{(i+k)}$ does not depend on j. □

3.2.2. Reordered tensor modes. In the definition below we introduce the type-1 and type-2 reordered tensor modes, formed by concatenating the type-1 and

type-2 oblique submatrices.

DEFINITION 3.6. *The type-*1 reordered tensor modes *are defined by concatenating the type-*1 *oblique submatrices:*

- A'_1 *is the* $I_1 \times (I_2 I_3)$ *matrix* $[T^{(1,3,2)}_{I_2-1}, \dots, T^{(1,3,2)}_0 = T^{(1,2,3)}_0, \dots, T^{(1,2,3)}_{I_3-1}]$,
- A'_2 *is the* $I_2 \times (I_3 I_1)$ *matrix* $[T^{(2,1,3)}_{I_3-1}, \dots, T^{(2,1,3)}_0 = T^{(2,3,1)}_0, \dots, T^{(2,3,1)}_{I_1-1}]$,
- A'_3 *is the* $I_3 \times (I_1 I_2)$ *matrix* $[T^{(3,2,1)}_{I_1-1}, \dots, T^{(3,2,1)}_0 = T^{(3,1,2)}_0, \dots, T^{(3,1,2)}_{I_2-1}]$.

*In the same way, the type-*2 reordered tensor modes *are defined by concatenating the type-*2 *oblique submatrices:*

- A''_1 *is the* $I_1 \times (I_2 I_3)$ *matrix* $[R^{(1,2,3)}_0, \dots, R^{(1,2,3)}_{I_2+I_3-2}]$,
- A''_2 *is the* $I_2 \times (I_3 I_1)$ *matrix* $[R^{(2,3,1)}_0, \dots, R^{(2,3,1)}_{I_3+I_1-2}]$,
- A''_3 *is the* $I_3 \times (I_1 I_2)$ *matrix* $[R^{(3,1,2)}_0, \dots, R^{(3,1,2)}_{I_1+I_2-2}]$.

PROPOSITION 3.7.

1. *For all* $s = 1, 2, 3$*, the mode* A_s *and the reordered modes* A'_s*,* A''_s *admit the same singular values and left singular vectors.*
2. *If* $\mathcal{A}$ *is a symmetric tensor, then* $A'_1 = A'_2 = A'_3$*, and this unique mode admits an axial blockwise symmetry w.r.t. its central oblique submatrix.*

Proof.

1. For all $s = 1, 2, 3$, the columns of the reordered modes A'_s and A''_s form a permutation of the columns of the mode A_s defined in section 2.
2. This is a corollary of point 2 in Proposition 3.5. ∎

4. Fast algorithms for computing the HOSVD of structured tensors. In this section, the reordered tensor modes introduced previously are used to efficiently compute the HOSVD of structured tensors. The first improvement consists of exploiting the column-redundancy of symmetric and Hankel tensors. To further reduce the computational cost, we then exploit the fast matrix-vector product techniques specific to Toeplitz and Hankel matrices.

4.1. Algorithms exploiting column-redundancy. Here we suppose that the s-mode of tensor $\mathcal{A}$ is redundant, e.g., some columns of the s-mode are equal (this is the case of symmetric and Hankel tensors, for example). We aim at exploiting this redundancy in order to efficiently implement the HOSVD of $\mathcal{A}$. Toward this end, we define the $I_s \times J_s$ matrix H_s as the matrix obtained by removing the repeated columns in the s-mode ($J_s \leq \prod_{s' \neq s} I_{s'}$), and we denote $d^{(s)}_k$ the number of occurrences of the kth column of H_s in the s-mode. Then we consider the $I_s \times I_s$ correlation matrix of the s-mode: $C^{(s)} = A_s {A_s}^H$. It is clear that this matrix can be factorized as

$$C^{(s)} = H_s D_s^2 H_s^H,$$

where

$$D_s = \operatorname{diag}\left(\sqrt{d^{(s)}_0} \dots \sqrt{d^{(s)}_{J_s-1}}\right)$$

(if the s-mode is not redundant, then we define H_s as the s-mode itself and D_s is defined as the $J_s \times J_s$ identity matrix). As a consequence, the M_s highest singular values and left singular vectors of the s-mode of dimensions $I_s \times \prod_{s' \neq s} I_{s'}$ are the same as those of the smaller $I_s \times J_s$ matrix $H_s D_s$.

Algorithms for symmetric tensors. In the case of $(I \times I \times I)$ symmetric tensors, we proved in point 2 of Proposition 3.7 that $A'_1 = A'_2 = A'_3$, and that this unique mode admits an axial blockwise symmetry. Therefore we can define the following:

- the nonredundant matrix $H_s = [T_0^{(1,2,3)}, \ldots, T_{I-1}^{(1,2,3)}]$, for all $s \in \{1,2,3\}$, of dimension $I \times J$ with $J = I(I+1)/2$;
- the weighting factors $d_k^{(s)} = \begin{cases} 1 & \text{if } 0 \le k < I, \\ 2 & \text{if } I \le k < J. \end{cases}$

In this way, the cost of the (rank-truncated) HOSVD is reduced to that of the (rank-truncated) SVD of $H_s D_s$, which is $2MI^3$ flops per iteration. In particular, it can be noted that the compression and weighting of the modes lead to a complexity six times as low as that of the algorithm in Table 2.1.

Algorithms for Hankel tensors. In the case of $(I_1 \times I_2 \times I_3)$ Hankel tensors, $[\mathcal{A}]_{i_1 i_2 i_3}$ is of the form $[\mathcal{A}]_{i_1 i_2 i_3} = x_{(i_1+i_2+i_3)}$, and we proved in point 3 of Proposition 3.5 that for all permutation π and index $k \in \{0, \ldots, I_{\pi_2} + I_{\pi_3} - 2\}$, $[R_k^{(\pi)}]_{ij} = x_{(i+k)}$. In particular, all columns of the type-2 oblique submatrix $R_k^{(\pi)}$ are equal. Therefore, for each s-mode we can define the following:

- the nonredundant Hankel matrix $H_s(i,k) = x_{(i+k)}$ of dimension $I_s \times J_s$ with $J_s = (\sum_{s' \neq s} I_{s'}) - 1$;
- the weighting factors $d_k^{(s)} = J^{(2)}_{\{\pi_{s'}\}_{s' \neq s}}(k) = \min(\{I_{s'}\}_{s' \neq s}, 1+k, J_s - k)$ (here $d_k^{(s)}$ is the number of columns of the kth oblique submatrix of the s-mode, defined in (3.5)). It can be noted that the weighting function $1 + k \mapsto d_k^{(s)}$ (plotted in Figure 4.1) is piecewise linear:

(4.1)
$$d_k^{(s)} = \begin{cases} 1+k & \text{if } 1 \le 1+k < \min(\{I_{s'}\}_{s' \neq s}), \\ \min(\{I_{s'}\}_{s' \neq s}) & \text{if } \min(\{I_{s'}\}_{s' \neq s}) \le 1+k \le \max(\{I_{s'}\}_{s' \neq s}), \\ J_s - k & \text{if } \max(\{I_{s'}\}_{s' \neq s}) < 1+k \le J_s, \\ 0 & \text{elsewhere.} \end{cases}$$

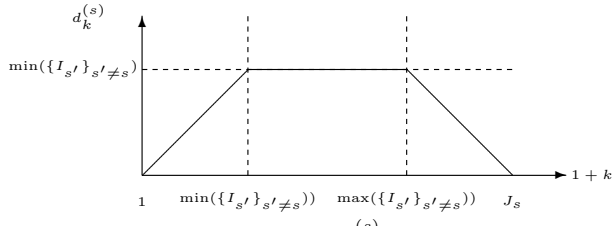


FIG. 4.1. *Weighting function $d_k^{(s)}$ for Hankel tensors.*

TABLE 4.1
Fast HOSVD algorithms for Hankel tensors.

(the cost corresponds to a single iteration of the orthogonal iteration method)

Operation	Cost per iteration
SVD of $H_1\, D_1$	$4M_1 I_1 (I_2 + I_3)$
SVD of $H_2\, D_2$	$4M_2 I_2 (I_1 + I_3)$
SVD of $H_3\, D_3$	$4M_3 I_3 (I_1 + I_2)$
Global cost	$24MI^2$

The fast SVD-based algorithm for computing the full or rank-truncated HOSVD of the Hankel tensor $\mathcal{A}$ is summarized in Table 4.1. The compression and weighting

TABLE 4.2
Fast HOSVD algorithm for Toeplitz tensors.

(the cost corresponds to a single iteration of the orthogonal iteration method)

Operation	Cost per iteration
SVD of A'_1	$2M_1(90I^2\log_2(I) + M_1I_1)$
SVD of A'_2	$2M_2(90I^2\log_2(I) + M_2I_2)$
SVD of A'_3	$2M_3(90I^2\log_2(I) + M_3I_3)$
Global cost	$6M\ 90I^2\log_2(I)$

of the modes allow a reduction of the complexity of one order of magnitude w.r.t. the algorithm in Table 2.1. If, additionally, the Hankel tensor is cube ($I_1 = I_2 = I_3 = I$), then it is symmetric and the three modes are equal. In this case, the global complexity is reduced $8MI^2$ flops.

4.2. Algorithms exploiting the Toeplitz or Hankel structure. In the above developments, we assumed that the rank-r rank-truncated SVD of an $n \times m$ matrix with $n < m$ was computed by means of the orthogonal iteration method [7, pp. 410–411], which consists of recursively performing $2r$ matrix/vector products and 1 QR factorization of an $n \times r$ matrix (a full SVD corresponds to the case $r = n$). We mentioned that the computational cost of one iteration is $2r\,c(n,m) + 2r^2n$ flops, where $c(n,m)$ is the cost of 1 matrix/vector product, and $2r^2n$ is the cost of 1 QR factorization [7, pp. 231–232].

In the following, we will focus on the HOSVD of Toeplitz or Hankel tensors, which can be computed efficiently using fast matrix/vector products. Indeed, the computational cost of a product between a $p \times q$ Toeplitz or Hankel matrix and a vector can be reduced from $2pq$ flops to $15(p+q)\log_2(p+q)$ flops by means of fast Fourier transforms (FFTs) [7, pp. 188-191,201–202].

Algorithms for Toeplitz tensors. In the case of Toeplitz tensors, we mentioned in point 2 of Proposition 3.5 that for all permutation π and index $k \in \{0, \dots, I_{\pi_3} - 1\}$, the type-1 oblique submatrix $T_k^{(\pi)}$ is Toeplitz. Therefore the oblique modes A'_s are formed of Toeplitz blocks. As a consequence, the computational cost of the multiplication of A'_s by a vector of appropriate dimension can be reduced from $2I^3$ flops to $90I^2\log_2(I)$ flops.[1] By introducing those fast products into the orthogonal iteration method, the cost of the (rank-truncated) SVD of A'_s is reduced to $2M_s(90I^2\log_2(I) + M_sI_s)$ per iteration.

The fast algorithm for computing the full or rank-truncated HOSVD of a Toeplitz tensor is summarized in Table 4.2. If, additionally, the tensor $\mathcal{A}$ is symmetric, then the three modes are equal. Moreover, as shown in section 4.1, the SVD of A'_s can be replaced by that of H_sD_s, where the $I \times \frac{I(I+1)}{2}$ matrix H_s is also block-Toeplitz. Therefore the cost of the SVD of H_sD_s is half that of the SVD of A'_s. As a consequence, the compression and weighting of the modes lead to a complexity six times as low as that of the fast HOSVD algorithm in Table 4.2.

[1] Under the constraint $I_1+I_2+I_3 = 3I$, the maximum cost is obtained for cube tensors ($I_1 = I_2 = I_3 = I$). Besides, left or right multiplying an $I \times k$ oblique submatrix $T_{I-k}^{(\pi)}$ by a vector of appropriate dimension normally involves $2Ik$ flops. This complexity is reduced to $15(I+k)\log_2(I+k)$ flops by means of FFT's. Therefore, left or right multiplying the block-Toeplitz matrix A'_s by a vector of appropriate dimension normally involves $2\sum_{k=0}^{I-1} 2Ik \sim 2I^3$ flops, or $2\sum_{k=0}^{I-1} 15(I+k)\log_2(I+k) \sim 90I^2\log_2(I)$ by means of FFT's.

Algorithms for Hankel tensors. In the case of Hankel tensors, we noted in section 4.1 that the HOSVD could be obtained by computing the SVD of the matrices $H_s D_s$, where each compressed mode H_s is a Hankel matrix ($H_s(i,k) = x_{(i+k)}$). Therefore, we can again use fast matrix-vector products to further reduce the complexity. More precisely, the computational cost of the multiplication of the $I_s \times ((\sum_{s' \neq s} I_{s'}) - 1)$ Hankel matrix H_s by a vector of appropriate dimension can be reduced from $4I^2$ flops to $45I \log_2(I)$ flops, by means of FFT's. By introducing those fast products into the orthogonal iteration method, the cost of the SVD of $H_s D_s$ is reduced to $2M_s(45I \log_2(I) + M_s I_s)$ per iteration.

The ultrafast algorithm for computing the full or rank-truncated HOSVD of a Hankel tensor is summarized in Table 4.3. Its global cost is provided as a maximum over M_1, M_2, M_3, under the constraint $M_1 + M_2 + M_3 = 3M$. It can be noted that the cost due to the fast matrix/vector products and the cost due to the QR factorizations can be of the same order of magnitude if $M = O(\log_2(I))$.

TABLE 4.3
Ultrafast HOSVD algorithm for Hankel tensors.

(the cost corresponds to a single iteration of the orthogonal iteration method)

Operation	Cost per iteration
SVD of $H_1 D_1$	$2M_1(45I \log_2(I) + M_1 I_1)$
SVD of $H_2 D_2$	$2M_2(45I \log_2(I) + M_2 I_2)$
SVD of $H_3 D_3$	$2M_3(45I \log_2(I) + M_3 I_3)$
Global cost	$6M(45I \log_2(I) + MI)$

If, additionally, the Hankel tensor is cube ($I_1 = I_2 = I_3 = I$), then it is symmetric, and the three modes are equal. In this case, the global complexity is three times as low as that of the ultrafast HOSVD algorithm in Table 4.3.

TABLE 4.4
Complexities of the HOSVD algorithms.

(the cost corresponds to a single iteration of the orthogonal iteration method)

Structure	Global cost per iteration
Unstructured	$12MI^3$
Symmetric	$2MI^3$
Toeplitz (fast)	$540MI^2 \log_2(I)$
Symmetric Toeplitz (fast)	$90MI^2 \log_2(I)$
Hankel (fast)	$24MI^2$
Cube Hankel (fast)	$8MI^2$
Hankel (ultrafast)	$270MI \log_2(I) + 6M^2 I$
Cube Hankel (ultrafast)	$90MI \log_2(I) + 2M^2 I$

4.3. Comparison of the complexities. The overall costs of the various HOSVD algorithms presented above are summarized in Table 4.4 (sorted in decreasing order of complexity). Note that only the complexity upper bounds are given in this table, and that the calculation of the tensor $\mathcal{S}$ is not included. Besides, it can be noted that the FFT-based HOSVD algorithms are not always the fastest, because of the high constants in Table 4.4. The best choice for computing the HOSVD actually depends on I and possibly on M (the dominant cost of all algorithms is linear w.r.t. M, except that of the ultrafast algorithms for Hankel tensors). Figure 4.2 represents the different complexities for $M = 10$. From this figure we can draw general remarks, which actually stand for any value of parameter M:

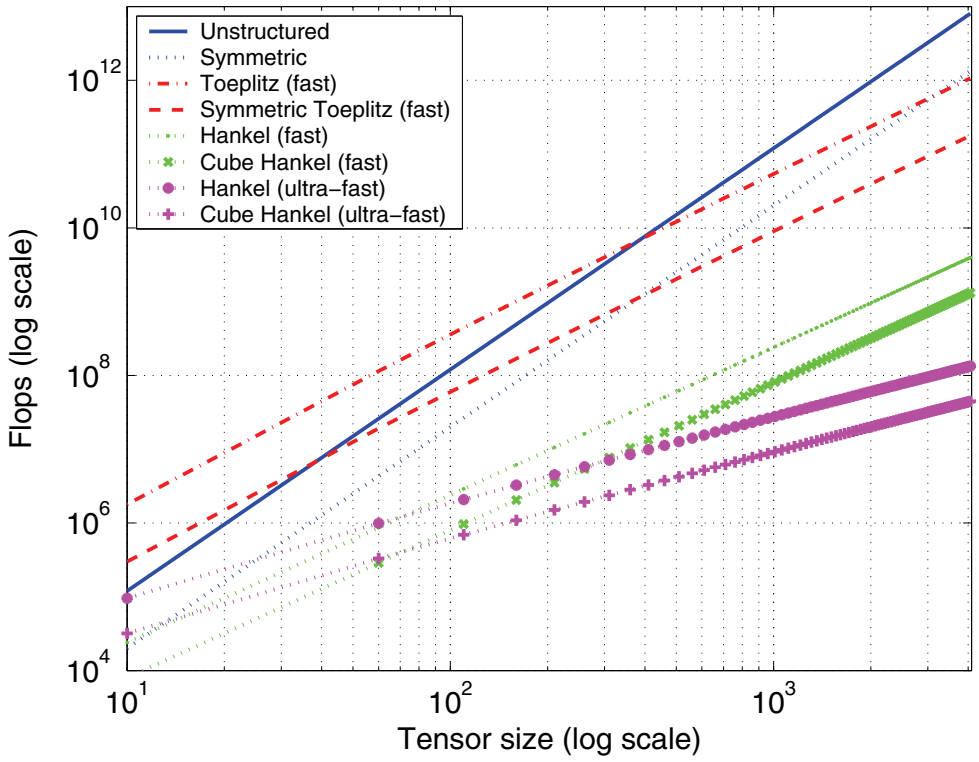


FIG. 4.2. *Flops count versus size I for $M = 10$.*

- the best algorithm for computing the HOSVD of Toeplitz (resp., symmetric Toeplitz) tensors is that dedicated to such tensors if $I \gtrsim 400$, or that dedicated to unstructured (resp., symmetric) tensors otherwise;
- the best algorithms for computing the HOSVD of symmetric, Hankel, and cube Hankel tensors are always those dedicated to such tensors.

In other respects, the comparison between the fast and ultrafast computations of the HOSVD for Hankel and cube Hankel tensors are sensitive to parameter M, as can be noted in Table 4.4. Our simulations showed the following:

- for small values of M ($M \ll I$), the ultrafast algorithm is faster if $I \gtrsim 70$;
- for moderate values of M ($M \simeq I/2$), the ultrafast algorithm is faster if $I \gtrsim 80$;
- for large values of M ($M \simeq I$), the ultrafast algorithm is faster if $I \gtrsim 100$.

5. Conclusions. In this paper, we proposed decreasing the computational cost of the full or rank-truncated HOSVD, which is basically $O(MI^3)$, by exploiting the structure of symmetric, Toeplitz, and Hankel tensors. For symmetric and Hankel tensors, our solution is based on the fact that the HOSVD can be reduced to the SVD of three nonredundant (no columns are repeated) matrices whose columns are multiplied by a given weighting function. In the case of Toeplitz and Hankel tensors, we propose a new way to perform the tensor unfolding which allows fast matrix/vector products. Finally, our fastest implementation of the HOSVD has a complexity of $O(MI\log_2(I))$ in the case of Hankel tensors.

REFERENCES

[1] C. A. Anderson and R. Bro, *Improving the speed of multi-way algorithms: Part* I. *Tucker* 3, Chemom. Intell. Lab. Syst., 42 (1998), pp. 93–103.

[2] M. Bebendorf, *Approximation of boundary element matrices*, Numer. Math., 86 (2000), pp. 565–589.

[3] R. Boyer and R. Badeau, *Adaptive multilinear SVD for structured tensors*, in Proceedings of ICASSP'06, vol. 3, Toulouse, France, IEEE, 2006, pp. 880–883.

[4] J. D. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[5] P. Comon, *Tensor Decompositions, state of the art and applications*, in IMA Conference on Mathematics in Signal Processing, Warwick, UK, 2000.

[6] L. De Lathauwer, *Signal Processing based on Multilinear Algebra*, Ph.D. thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 1997.

[7] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

[8] R. A. Harshman and M. E. Lundy, *PARAFAC: Parallel factor analysis*, Comp. Stat. Data Anal., 18 (1994), pp. 39–72.

[9] T. D. Howell, *Global properties of tensor rank*, Linear Algebra Appl., 22 (1978), pp. 9–23.

[10] J. Huang, H. Wium, K. B. Qvist, and K. Esbensen, *Multi-way methods in image analysis—relationships and applications*, Chemom. Intell. Lab. Syst., 66 (2003), pp. 141–158.

[11] T. Kailath and A. H. Sayed, eds., *Fast Reliable Algorithms for Matrices with Structure*, SIAM, Philadelphia, 1999.

[12] H. A. Kiers, *Towards a standardized notation and terminology in multiway analysis*, J. Chemometrics, 14 (2000), pp. 105–122.

[13] P. M. Kroonenberg, *Three-Mode Principal Component Analysis: Theory and Applications*, DSWO Press, Leiden University, Faculty of Social and Behavioural Sciences, Department of Data Theory, Leiden, The Netherlands, 1983.

[14] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[15] X. Liu and N.D. Sidiropoulos, *Almost sure identifiability of constant modulus multidimensional harmonic retrieval*, IEEE Trans. Signal. Process., 50 (2002), pp. 2366–2368.

[16] I. V. Oseledets, D. V. Savostianov, and E. E. Tyrtyshnikov, *Tucker dimensionality reduction in linear time*, SIAM J. Matrix Anal., submitted.

[17] J. M. Papy, L. De Lathauwer, and S. Van Huffel, *Exponential data fitting using multilinear algebra: The single-channel and the multichannel case*, Numer. Linear Algebra Appl., 12 (2005), pp. 809–826.

[18] B. Savas, *Analyses and Tests of Handwritten Digit Recognition Algorithms*, Master's thesis, Department of Mathematics, Linkoping University, Linkoping, Sweden, 2003.

[19] N. D. Sidiropoulos, *Low-Rank Decomposition of Multi-Way Arrays: A Signal Processing Perspective*, in IEEE Workshop on Sensor Array and Multichannel Processing (SAM2004), Barcelona, Spain, 2004.

[20] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen, *CubeSVD: A Novel Approach to Personalized Web Search*, in Proceedings of the International World Wide Web Conference, Chiba, Japan, 2005, pp. 382–390.

[21] L. R. Tucker, *Some Mathematical Notes on Three-mode Factor Analysis*, Psychometrika, 31 (1996), pp. 279–311.

[22] E. E. Tyrtyshnikov, *Incomplete cross approximation in the mosaic–skeleton method*, Computing, 64 (2000), pp. 367–380.

[23] H. Wang and N. Ahuja, *Facial Expression Decomposition*, in Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV), Nice, France, 2003, Vol. 2, pp. 958–965.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1022–1032

DECOMPOSITIONS OF A HIGHER-ORDER TENSOR IN BLOCK TERMS—PART I: LEMMAS FOR PARTITIONED MATRICES*

LIEVEN DE LATHAUWER†

Abstract. In this paper we study a generalization of Kruskal's permutation lemma to partitioned matrices. We define the k'-rank of partitioned matrices as a generalization of the k-rank of matrices. We derive a lower-bound on the k'-rank of Khatri–Rao products of partitioned matrices. We prove that Khatri–Rao products of partitioned matrices are generically full column rank.

Key words. multilinear algebra, higher-order tensor, Tucker decomposition, canonical decomposition, parallel factors model

AMS subject classifications. 15A18, 15A69

DOI. 10.1137/060661685

1. Introduction.

1.1. Organization of the paper. In a companion paper we introduce decompositions of a higher-order tensor in several types of block terms [3]. For the analysis of these decompositions, we need a number of tools. Some of these are introduced in the present paper. In section 2 we derive a generalization of Kruskal's permutation lemma [6], which we call the equivalence lemma for partitioned matrices. Section 2 also introduces the k'-rank of partitioned matrices as a generalization of the k-rank of matrices [6]. In section 3 we present some results on the rank and k'-rank of Khatri–Rao products of partitioned matrices (see (1.1)).

1.2. Notation. We use $\mathbb{K}$ to denote $\mathbb{R}$ or $\mathbb{C}$ when the difference is not important. In this paper scalars are denoted by lowercase letters (a, b, ...), vectors are written in boldface lowercase ($\mathbf{a}$, $\mathbf{b}$, ...), and matrices correspond to boldface capitals ($\mathbf{A}$, $\mathbf{B}$, ...). This notation is consistently used for lower-order parts of a given structure. For instance, the entry with row index i and column index j in a matrix $\mathbf{A}$, i.e., $(\mathbf{A})_{ij}$, is symbolized by a_{ij} (also $(\mathbf{a})_i = a_i$). If no confusion is possible, the ith column vector of a matrix $\mathbf{A}$ is denoted as $\mathbf{a}_i$, i.e., $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \ldots]$. Sometimes we use the MATLAB colon notation to indicate submatrices of a given matrix or subtensors of a given tensor. Italic capitals are also used to denote index upper bounds (e.g., $i = 1, 2, \ldots, I$). The symbol $\otimes$ denotes the Kronecker product,

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \ldots \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \ldots \\ \vdots & \vdots & \end{pmatrix}.$$

*Received by the editors June 1, 2006; accepted for publication (in revised form) by J. G. Nagy April 14, 2008; published electronically September 25, 2008. This research was supported by Research Council K.U.Leuven: GOA-Ambiorics, CoE EF/05/006 Optimization in Engineering (OPTEC), CIF1; F.W.O.: project G.0321.06 and Research Communities ICCoS, ANMMM, and MLDM; the Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, "Dynamical systems, control and optimization," 2007–2011); and the EU: ERNSI.

http://www.siam.org/journals/simax/30-3/66168.html

†Subfaculty Science and Technology, Katholieke Universiteit Leuven Campus Kortrijk, E. Sabbelaan 53, 8500 Kortrijk, Belgium (Lieven.DeLathauwer@kuleuven-kortrijk.be), and Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (Lieven.DeLathauwer@esat.kuleuven.be, http://homes.esat.kuleuven.be/~delathau/home.html).

Let $\mathbf{A} = [\mathbf{A}_1 \ \dots \ \mathbf{A}_R]$ and $\mathbf{B} = [\mathbf{B}_1 \ \dots \ \mathbf{B}_R]$ be two partitioned matrices. Then the Khatri–Rao product is defined as the partitionwise Kronecker product and represented by $\odot$ [7]:

$$\mathbf{A} \odot \mathbf{B} = (\mathbf{A}_1 \otimes \mathbf{B}_1 \dots \mathbf{A}_R \otimes \mathbf{B}_R)\,. \tag{1.1}$$

In recent years, the term "Khatri–Rao product" and the symbol $\odot$ have been used mainly in cases where $\mathbf{A}$ and $\mathbf{B}$ are partitioned into vectors. For clarity, we denote this particular, columnwise Khatri–Rao product by $\odot_c$:

$$\mathbf{A} \odot_c \mathbf{B} = (\mathbf{a}_1 \otimes \mathbf{b}_1 \dots \mathbf{a}_R \otimes \mathbf{b}_R)\,.$$

The column space of a matrix and its orthogonal complement will be denoted by $\mathrm{span}(\mathbf{A})$ and $\mathrm{null}(\mathbf{A})$. The rank of a matrix $\mathbf{A}$ will be denoted by $\mathrm{rank}(\mathbf{A})$ or $r_{\mathbf{A}}$. The superscripts $\cdot^T$, $\cdot^H$, and $\cdot^\dagger$ denote the transpose, complex conjugated transpose, and Moore–Penrose pseudoinverse, respectively. The $(N \times N)$ identity matrix is represented by $\mathbf{I}_{N\times N}$. The $(I \times J)$ zero matrix is denoted by $\mathbf{0}_{I\times J}$.

2. The equivalence lemma for partitioned matrices. Let $\omega(\mathbf{x})$ denote the number of nonzero entries of a vector $\mathbf{x}$. The following lemma was originally proposed by Kruskal in [6]. It is known as the *permutation lemma*. It plays a crucial role in the analysis of the uniqueness of the canonical/parallel factor (CANDECOMP/PARAFAC) decomposition [1, 5]. The proof was reformulated in terms of accessible basic linear algebra in [9]. An alternative proof was given in [4]. The link between the two proofs is also discussed in [9].

LEMMA 2.1 (permutation lemma). *Consider two matrices $\bar{\mathbf{A}}, \mathbf{A} \in \mathbb{K}^{I\times R}$ that have no zero columns. If for every vector $\mathbf{x}$ such that $\omega(\mathbf{x}^T\bar{\mathbf{A}}) \leqslant R - r_{\bar{\mathbf{A}}} + 1$, we have $\omega(\mathbf{x}^T\mathbf{A}) \leqslant \omega(\mathbf{x}^T\bar{\mathbf{A}})$, then there exists a unique permutation matrix $\mathbf{\Pi}$ and a unique nonsingular diagonal matrix $\mathbf{\Lambda}$ such that $\bar{\mathbf{A}} = \mathbf{A}\cdot\mathbf{\Pi}\cdot\mathbf{\Lambda}$.*

Below, we present a generalization of the permutation lemma for matrices that are partitioned as in $\mathbf{A} = [\mathbf{A}_1 \dots \mathbf{A}_R]$. This generalization is essential in the study of the uniqueness of the decompositions introduced in [3].

Let us first introduce some additional prerequisites. Let $\omega'(\mathbf{x})$ denote the number of parts of a partitioned vector $\mathbf{x}$ that are not all-zero. We call the partitioning of a partitioned matrix $\mathbf{A}$ uniform when all submatrices are of the same size. We also have the following definition.

DEFINITION 2.2. *The* Kruskal rank *or* k-rank *of a matrix $\mathbf{A}$, denoted by* $\mathrm{rank}_k(\mathbf{A})$ *or $k_{\mathbf{A}}$, is the maximal number r such that any set of r columns of $\mathbf{A}$ is linearly independent* [6].

We call a property generic when it holds with probability one when the parameters of the problem are drawn from continuous probability density functions. Let $\mathbf{A} \in \mathbb{K}^{I\times R}$. Generically, we have $k_{\mathbf{A}} = \min(I, R)$. K-ranks appear in the formulation of the famous Kruskal condition for CANDECOMP/PARAFAC uniqueness (see [3, Theorem 1.14]).

We now generalize the k-rank concept to partitioned matrices.

DEFINITION 2.3. *The* k'-rank *of a (not necessarily uniformly) partitioned matrix $\mathbf{A}$, denoted by* $\mathrm{rank}_{k'}(\mathbf{A})$ *or $k'_{\mathbf{A}}$, is the maximal number r such that any set of r submatrices of $\mathbf{A}$ yields a set of linearly independent columns.*

Let $\mathbf{A} \in \mathbb{K}^{I\times LR}$ be uniformly partitioned in R matrices $\mathbf{A}_r \in \mathbb{K}^{I\times L}$. Generically, we have $k'_{\mathbf{A}} = \min(\lfloor \frac{I}{L} \rfloor, R)$. K'-ranks will appear in the formulation of generalizations of Kruskal's condition to block term decompositions [3].

The generalization of the permutation lemma to partitioned matrices is now as follows.

LEMMA 2.4 (equivalence lemma for partitioned matrices). *Consider* $\bar{\mathbf{A}}, \mathbf{A} \in \mathbb{K}^{I \times \sum_{r=1}^{R} L_r}$*, partitioned in the same but not necessarily uniform way into* R *submatrices that are full column rank. Suppose that for every* $\mu \leqslant R - k'_{\bar{\mathbf{A}}} + 1$ *there holds that for a generic*[1] *vector* $\mathbf{x}$ *such that* $\omega'(\mathbf{x}^H \bar{\mathbf{A}}) \leqslant \mu$*, we have* $\omega'(\mathbf{x}^H \mathbf{A}) \leqslant \omega'(\mathbf{x}^H \bar{\mathbf{A}})$*. Then there exists a unique block-permutation matrix* $\mathbf{\Pi}$ *and a unique nonsingular block-diagonal matrix* $\mathbf{\Lambda}$*, such that* $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}$*, where the block-transformation is compatible with the block-structure of* $\mathbf{A}$ *and* $\bar{\mathbf{A}}$*.*

The permutation lemma is not only about permutations. Rather it gives a condition under which two matrices are *equivalent* up to columnwise permutation *and scaling*. The lemma thus makes sure that two matrices belong to the same quotient class of the equivalence relation defined by $\mathbf{A} \sim \mathbf{B} \Leftrightarrow \mathbf{A} = \mathbf{B} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}$, in which $\mathbf{\Pi}$ is an arbitrary permutation matrix and $\mathbf{\Lambda}$ an arbitrary nonsingular diagonal matrix, respectively. We find it therefore appropriate to call Lemma 2.4 the *equivalence* lemma for partitioned matrices.

We note that the rank $r_{\bar{\mathbf{A}}}$ in the permutation lemma has been replaced by the k'-rank $k'_{\bar{\mathbf{A}}}$ in Lemma 2.4, because the permutation lemma admits a simpler proof when we can assume that $r_{\bar{\mathbf{A}}} = k_{\bar{\mathbf{A}}}$. It is this simpler proof, given in [4], that will be generalized in this paper. We stay quite close to the text of [4]. We recommend studying the proof in [4] before reading the remainder of this section.

We work as follows. First we have a closer look at the meaning of the condition in the equivalence lemma for partitioned matrices (Lemma 2.5). Then we prove that $\mathbf{A}$ and $\bar{\mathbf{A}}$ are equivalent when the condition in the equivalence lemma for partitioned matrices holds for all $\mu \leqslant R$ (Lemma 2.6). Finally we show that it is sufficient to claim that the condition holds for $\mu \leqslant R - k'_{\bar{\mathbf{A}}} + 1$ (Lemma 2.7).

LEMMA 2.5. *Consider* $\bar{\mathbf{A}}, \mathbf{A} \in \mathbb{K}^{I \times L}$*, partitioned in the same but not necessarily uniform way into* R *submatrices that are full column rank. The following two statements are equivalent:*

(i) *For every* $\mu \leqslant R - k'_{\bar{\mathbf{A}}} + 1$ *there holds that for a generic vector* $\mathbf{x}$ *such that* $\omega'(\mathbf{x}^H \bar{\mathbf{A}}) \leqslant \mu$*, we have* $\omega'(\mathbf{x}^H \mathbf{A}) \leqslant \omega'(\mathbf{x}^H \bar{\mathbf{A}})$*.*

(ii) *If a vector is orthogonal to* $c \geqslant k'_{\bar{\mathbf{A}}} - 1$ *submatrices of* $\bar{\mathbf{A}}$*, then it must generically be orthogonal to at least* c *submatrices of* $\mathbf{A}$*.*

These, in turn, imply the following:

(iii) *For every set of* $c \geqslant k'_{\bar{\mathbf{A}}} - 1$ *submatrices of* $\bar{\mathbf{A}}$*, there exists a set of at least* c *submatrices of* $\mathbf{A}$ *such that span(matrix formed by these* $c \geqslant k'_{\bar{\mathbf{A}}} - 1$ *submatrices of* $\bar{\mathbf{A}}$*)* $\supseteq$ *span(matrix formed by the* c *or more submatrices of* $\mathbf{A}$*).*

Proof. The equivalence of (i) and (ii) follows directly from the definition of $\omega'(\mathbf{x})$.

[1] We mean the following. Consider, for instance, a partitioned matrix $\bar{\mathbf{A}} = [\mathbf{a}_1\ \mathbf{a}_2 | \mathbf{a}_3\ \mathbf{a}_4] \in \mathbb{K}^{4 \times 4}$ that is full column rank. The set $\mathsf{S} = \{\mathbf{x} | \omega'(\mathbf{x}^H \bar{\mathbf{A}}) \leqslant 1\}$ is the union of two subspaces, S_1 and S_2, consisting of the set of vectors orthogonal to $\{\mathbf{a}_1, \mathbf{a}_2\}$ and $\{\mathbf{a}_3, \mathbf{a}_4\}$, respectively. When we say that for a generic vector $\mathbf{x}$ such that $\omega'(\mathbf{x}^H \bar{\mathbf{A}}) \leqslant 1$, we have $\omega'(\mathbf{x}^H \mathbf{A}) \leqslant \omega'(\mathbf{x}^H \bar{\mathbf{A}})$, we mean that $\omega'(\mathbf{x}^H \mathbf{A}) \leqslant \omega'(\mathbf{x}^H \bar{\mathbf{A}})$ holds with probability one for a vector $\mathbf{x}$ drawn from a continuous probability density function over S_1 and that $\omega'(\mathbf{x}^H \mathbf{A}) \leqslant \omega'(\mathbf{x}^T \bar{\mathbf{A}})$ also holds with probability one for a vector $\mathbf{x}$ drawn from a continuous probability density function over S_2. In general, the set $\mathsf{S} = \{\mathbf{x} | \omega'(\mathbf{x}^H \bar{\mathbf{A}}) \leqslant \mu\}$ consists of a finite union of subspaces, where we count only the subspaces that are not contained in another subspace. For each of these subspaces, the property should hold with probability one for a vector $\mathbf{x}$ drawn from a continuous probability density function over that subspace.

We now prove in two ways that (ii) implies (iii). The first proof is a generalization of [4, Remark 1]. This proof is by contradiction. Suppose that there is a set of $c_0 \geqslant k'_{\bar{\mathbf{A}}} - 1$ submatrices of $\bar{\mathbf{A}}$, say, $\bar{\mathbf{A}}_1, \ldots, \bar{\mathbf{A}}_{c_0}$, and that there are only $c_0 - k$ submatrices of $\mathbf{A}$, say, $\mathbf{A}_1, \ldots, \mathbf{A}_{c_0-k}$, such that

$$\operatorname{span}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{c_0}]) \supseteq \operatorname{span}([\mathbf{A}_1 \ \ldots \ \mathbf{A}_{c_0-k}]),$$

where $1 \leqslant k \leqslant c_0$. The column space of none of the remaining submatrices of $\mathbf{A}$, i.e., $\mathbf{A}_{c_0-k+1}, \ldots, \mathbf{A}_R$, is contained in $\operatorname{span}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{c_0}])$; otherwise, k can be reduced. This implies that for every $i = c_0 - k + 1, \ldots, R$, there exists a certain nonzero vector $\mathbf{x}_i \in \operatorname{null}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{c_0}])$ such that

$$\mathbf{x}_i^H \mathbf{A}_i \neq [0 \ \ldots \ 0]. \tag{2.1}$$

We can assume that $\operatorname{null}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{c_0}])$ is a subspace of dimension $m \geqslant 1$. The case $m = 0$ corresponds to $\operatorname{span}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{c_0}]) = \mathbb{K}^I$. In this case, the span of all submatrices of $\mathbf{A}$ is contained in $\operatorname{span}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{c_0}])$.

Due to the existence of $\mathbf{x}_i$ in (2.1), we have for $i = c_0 - k + 1, \ldots, R$ that $\operatorname{null}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{c_0}\mathbf{A}_i])$ is a proper subspace of $\operatorname{null}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{c_0}])$ with dimension at most $m-1$. Since the union of a countable number of at most $(m-1)$-dimensional subspaces of $\mathbb{K}^I$ cannot cover an m-dimensional subspace of $\mathbb{K}^I$, there holds for a generic vector $\mathbf{x}_0 \in \operatorname{null}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{c_0}])$ that

$$\mathbf{x}_0^H \mathbf{A}_i \neq [0 \ \ldots \ 0], \quad i = c_0 - k + 1, \ldots, R.$$

We have a contradiction with (ii).

The second proof is direct.[2] If a vector is orthogonal to c submatrices of $\bar{\mathbf{A}}$, then it is in the left null space of c submatrices of $\bar{\mathbf{A}}$. Denote the matrix formed by these c submatrices by $\bar{\mathbf{A}}_c$. By assumption, we have that the vector is generically also in the left null space of $\bar{c} \geqslant c$ submatrices of $\mathbf{A}$. Denote the matrix formed by these $\bar{c}$ submatrices by $\mathbf{A}_{\bar{c}}$. Since

$$\operatorname{null}(\bar{\mathbf{A}}_c) \subseteq \operatorname{null}(\mathbf{A}_{\bar{c}})$$

we have

$$\operatorname{span}(\bar{\mathbf{A}}_c) \supseteq \operatorname{span}(\mathbf{A}_{\bar{c}}).$$

This completes the proof. ∎

We now demonstrate the equivalence of matrices under a condition that seems stronger than the one in the equivalence lemma for partitioned matrices.

LEMMA 2.6. *Consider $\bar{\mathbf{A}}, \mathbf{A} \in \mathbb{K}^{I \times L}$, partitioned in the same but not necessarily uniform way into R submatrices that are full column rank. The following two statements are equivalent:*

(i) *There exists a unique block-permutation matrix $\mathbf{\Pi}$ and a unique nonsingular block-diagonal matrix $\mathbf{\Lambda}$, such that $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}$, where the block-transformation is compatible with the block-structure of $\mathbf{A}$ and $\bar{\mathbf{A}}$.*

(ii) *For every $\mu \leqslant R$ there holds that, for a generic vector $\mathbf{x}$ such that $\omega'(\mathbf{x}^H \bar{\mathbf{A}}) \leqslant \mu$, we have $\omega'(\mathbf{x}^H \mathbf{A}) \leqslant \omega'(\mathbf{x}^H \bar{\mathbf{A}})$.*

[2] This proof was suggested by an anonymous reviewer.

Proof. The implication of (ii) from (i) is trivial. The implication of (i) from (ii) is proved by induction on the number of submatrices R.

For $R = 1$, the condition in the lemma means that $\omega'(\mathbf{x}^H\mathbf{A}) = 0$ for a generic vector $\mathbf{x}$ satisfying $\omega'(\mathbf{x}^H\bar{\mathbf{A}}) = 0$. This implies that $\text{null}(\bar{\mathbf{A}}) \subseteq \text{null}(\mathbf{A})$. Since $\text{null}(\mathbf{A})$ and $\text{null}(\bar{\mathbf{A}})$ are the orthogonal complements of $\text{span}(\mathbf{A})$ and $\text{span}(\bar{\mathbf{A}})$, respectively, we have $\text{span}(\mathbf{A}) \subseteq \text{span}(\bar{\mathbf{A}})$. Since both $\mathbf{A}$ and $\bar{\mathbf{A}}$ are full column rank, the dimensions of $\text{span}(\mathbf{A})$ and $\text{span}(\bar{\mathbf{A}})$ are equal. Hence, we have $\text{span}(\mathbf{A}) = \text{span}(\bar{\mathbf{A}})$ and $\mathbf{A} = \bar{\mathbf{A}} \cdot \mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is $(L \times L)$ nonsingular.

Now assume that the lemma holds for all $R \leqslant K$. We show that it then also holds for $R = K + 1$. The proof is by contradiction. We assume that in the induction step matrices $\mathbf{A}_1$ and $\bar{\mathbf{A}}_1$ are appended to $[\mathbf{A}_2 \ \ldots \ \mathbf{A}_{K+1}]$ and $[\bar{\mathbf{A}}_2 \ \ldots \ \bar{\mathbf{A}}_{K+1}]$, respectively. Both $\mathbf{A}_1$ and $\bar{\mathbf{A}}_1$ have L_1 columns. Without loss of generality, we assume that none of the other submatrices $\mathbf{A}_2, \ldots, \mathbf{A}_{K+1}, \bar{\mathbf{A}}_2, \ldots, \bar{\mathbf{A}}_{K+1}$ has less than L_1 columns.

Assume that $\text{span}(\bar{\mathbf{A}}_1)$ does not coincide with $\text{span}(\mathbf{A}_j)$ for any $j = 1, \ldots, R = K+1$. This means that for all j, $\text{span}([\bar{\mathbf{A}}_1 \ \mathbf{A}_j]) \supset \text{span}(\bar{\mathbf{A}}_1)$. Equivalently, $\text{null}(\bar{\mathbf{A}}_1) \supset \text{null}([\bar{\mathbf{A}}_1 \ \mathbf{A}_j])$. Denote $\dim(\text{null}(\bar{\mathbf{A}}_1)) = I - \alpha$ and $\dim(\text{null}([\bar{\mathbf{A}}_1 \ \mathbf{A}_j])) = I - \alpha - \beta_j$, with $\beta_j \geqslant 1$, $j = 1, \ldots, R$. Since the union of a countable number of subspaces of dimension $I - \alpha - \beta_j$ cannot cover a subspace of dimension $I - \alpha$, $\bigcup_{j=1}^{R} \text{null}([\bar{\mathbf{A}}_1 \ \mathbf{A}_j])$ does not cover $\text{null}(\bar{\mathbf{A}}_1)$. This implies that for a generic vector $\mathbf{x}_0$ in $\text{null}(\bar{\mathbf{A}}_1)$ we have

$$\omega'(\mathbf{x}_0^H\bar{\mathbf{A}}_1) = 0, \quad \omega'(\mathbf{x}_0^H\mathbf{A}_j) = 1, \quad j = 1, \ldots, R.$$

This means that for a generic vector $\mathbf{x}_0$ in $\text{null}(\bar{\mathbf{A}}_1)$ we have

$$\omega'(\mathbf{x}_0^H\bar{\mathbf{A}}) \leqslant R - 1 \leqslant R = \omega'(\mathbf{x}_0^H\mathbf{A}).$$

We have a contradiction with the condition in the lemma. Therefore, there exists a submatrix of $\mathbf{A}$, say, $\mathbf{A}_{j_0}$, such that $\bar{\mathbf{A}}_1 = \mathbf{A}_{j_0} \cdot \mathbf{L}$, in which $\mathbf{L}$ is square nonsingular.

We now construct a submatrix $\bar{\mathbf{A}}_0$ of $\bar{\mathbf{A}}$ by removing $\bar{\mathbf{A}}_1$ and a submatrix $\mathbf{A}_0$ of $\mathbf{A}$ by removing $\mathbf{A}_{j_0}$. Since for every vector $\mathbf{x}$, $\omega'(\mathbf{x}^H\bar{\mathbf{A}}_1) = \omega'(\mathbf{x}^H\mathbf{A}_{j_0})$ and, on the other hand, $\omega'(\mathbf{x}^H\mathbf{A}) \leqslant \omega'(\mathbf{x}^H\bar{\mathbf{A}})$ generically, we also have $\omega'(\mathbf{x}^H\mathbf{A}_0) \leqslant \omega'(\mathbf{x}^H\bar{\mathbf{A}}_0)$ generically. That is, $\mathbf{A}_0$ and $\bar{\mathbf{A}}_0$ satisfy the condition in the lemma, but they consist of only K submatrices. From the induction step we then have that $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}$. This completes the proof. ∎

As mentioned above, the condition in Lemma 2.6 can be relaxed to the one in the equivalence lemma for partitioned matrices.

LEMMA 2.7. *Consider $\bar{\mathbf{A}}, \mathbf{A} \in \mathbb{K}^{I \times L}$, partitioned in the same but not necessarily uniform way into R submatrices that are full column rank. The following two statements are equivalent*:

(i) *For every $\mu \leqslant R$ there holds that for a generic vector $\mathbf{x}$ such that $\omega'(\mathbf{x}^H\bar{\mathbf{A}}) \leqslant \mu$, we have $\omega'(\mathbf{x}^H\mathbf{A}) \leqslant \omega'(\mathbf{x}^H\bar{\mathbf{A}})$.*

(ii) *For every $\mu \leqslant R - k'_{\bar{\mathbf{A}}} + 1$ there holds that for a generic vector $\mathbf{x}$ such that $\omega'(\mathbf{x}^H\bar{\mathbf{A}}) \leqslant \mu$, we have $\omega'(\mathbf{x}^H\mathbf{A}) \leqslant \omega'(\mathbf{x}^H\bar{\mathbf{A}})$.*

Proof. The implication of (ii) from (i) is trivial. The implication of (i) from (ii) is proved by contradiction.

Suppose there exists a nonzero vector $\mathbf{x}_0$ such that $\omega'(\mathbf{x}_0^H\mathbf{A}) > \omega'(\mathbf{x}_0^H\bar{\mathbf{A}})$ while $\omega'(\mathbf{x}_0^H\bar{\mathbf{A}}) > R - k'_{\bar{\mathbf{A}}} + 1$. Suppose that $\omega'(\mathbf{x}_0^H\bar{\mathbf{A}})$ is the smallest number bigger than $R - k'_{\bar{\mathbf{A}}} + 1$ for which (ii) does not hold, i.e., suppose that for every $\mu < \omega'(\mathbf{x}_0^H\bar{\mathbf{A}})$ there holds that for a generic vector $\mathbf{x}$ such that $\omega'(\mathbf{x}^H\bar{\mathbf{A}}) \leqslant \mu$, we have $\omega'(\mathbf{x}^H\mathbf{A}) \leqslant \omega'(\mathbf{x}^H\bar{\mathbf{A}})$. We can write

$$\omega'(\mathbf{x}_0^H\bar{\mathbf{A}}) = R - k'_{\bar{\mathbf{A}}} + \alpha \tag{2.2}$$

with $2 \leqslant \alpha < k'_{\bar{\mathbf{A}}}$ and

$$\omega'(\mathbf{x}_0^H \mathbf{A}) = R - k'_{\bar{\mathbf{A}}} + \alpha + \beta \tag{2.3}$$

with $1 \leqslant \beta < k'_{\bar{\mathbf{A}}} - \alpha$. Associated with $\mathbf{x}_0$, we have $k'_{\bar{\mathbf{A}}} - \alpha$ submatrices of $\bar{\mathbf{A}}$, say, $\bar{\mathbf{A}}_1, \ldots, \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha}$, and $k'_{\bar{\mathbf{A}}} - \alpha - \beta$ submatrices of $\mathbf{A}$, say, $\mathbf{A}_1, \ldots, \mathbf{A}_{k'_{\bar{\mathbf{A}}}-\alpha-\beta}$, such that

$$\mathbf{x}_0 \in \operatorname{null}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha}]) \cap \operatorname{null}([\mathbf{A}_1 \ \ldots \ \mathbf{A}_{k'_{\bar{\mathbf{A}}}-\alpha-\beta}]).$$

$\mathbf{A}_1, \ldots, \mathbf{A}_{k'_{\bar{\mathbf{A}}}-\alpha-\beta}$ are the only submatrices of $\mathbf{A}$ of which the column space can possibly be contained in $\operatorname{span}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha}])$. Otherwise, if there is one more submatrix, say, $\mathbf{A}_R$, of which the column space is contained in $\operatorname{span}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha}])$, then $\mathbf{x}_0^H \mathbf{A}_R = \mathbf{0}$ such that $\omega'(\mathbf{x}_0^H \mathbf{A}) = R - k'_{\bar{\mathbf{A}}} + \alpha + \beta - 1$, which contradicts (2.3).

Recall that by definition of $\omega'(\mathbf{x}_0^H \bar{\mathbf{A}})$ for every $\mu \leqslant R - k'_{\bar{\mathbf{A}}} + \alpha - 1 < \omega'(\mathbf{x}_0^H \bar{\mathbf{A}})$ there holds that for generic $\mathbf{x}$ such that $\omega'(\mathbf{x}^H \bar{\mathbf{A}}) \leqslant \mu$, we have $\omega'(\mathbf{x}^H \mathbf{A}) \leqslant \omega'(\mathbf{x}^H \bar{\mathbf{A}})$. Similar to Lemma 2.5, we can show that this implies that for every set of $c \geqslant k'_{\bar{\mathbf{A}}} - \alpha + 1$ submatrices of $\bar{\mathbf{A}}$, there exists a set of at least c submatrices of $\mathbf{A}$ such that span(matrix formed by these $c \geqslant k'_{\bar{\mathbf{A}}} - \alpha + 1$ submatrices of $\bar{\mathbf{A}}$) $\supseteq$ span(matrix formed by the c or more submatrices of $\mathbf{A}$).

Now we consider the matrices $[\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha}]$ and $[\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha} \ \bar{\mathbf{A}}_i]$, $i = k'_{\bar{\mathbf{A}}} - \alpha + 1, \ldots, R$. For each of these matrices we consider the submatrices of $\mathbf{A}$ of which the column space is contained in the column space of the given matrix.

First, recall that $\mathbf{A}_1, \ldots, \mathbf{A}_{k'_{\bar{\mathbf{A}}}-\alpha-\beta}$ are the only submatrices of $\mathbf{A}$ of which the column space is contained in $\operatorname{span}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha}])$. Next, since $[\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha} \ \bar{\mathbf{A}}_i]$ consists of $k'_{\bar{\mathbf{A}}} - \alpha + 1$ submatrices of $\bar{\mathbf{A}}$, there exist at least $k'_{\bar{\mathbf{A}}} - \alpha + 1$ submatrices $\mathbf{A}_{i_1}, \ldots, \mathbf{A}_{i_{k'_{\bar{\mathbf{A}}}-\alpha+1}}$ such that $\operatorname{span}([\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha} \ \bar{\mathbf{A}}_i]) \supseteq \operatorname{span}([\mathbf{A}_{i_1} \ \ldots \ \mathbf{A}_{i_{k'_{\bar{\mathbf{A}}}-\alpha+1}}])$. Combining these results, we conclude that at least $\beta+1 = (k'_{\bar{\mathbf{A}}}-\alpha+1)-(k'_{\bar{\mathbf{A}}}-\alpha-\beta)$ submatrices of $[\mathbf{A}_{i_1} \ \ldots \ \mathbf{A}_{i_{k'_{\bar{\mathbf{A}}}-\alpha+1}}]$, other than $\mathbf{A}_1, \ldots, \mathbf{A}_{k'_{\bar{\mathbf{A}}}-\alpha-\beta}$, have a column space that is in the span of $[\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha} \ \bar{\mathbf{A}}_i]$. Denote by ϕ_i the set of those $\beta+1$ or more submatrices of $[\mathbf{A}_{i_1} \ \ldots \ \mathbf{A}_{i_{k'_{\bar{\mathbf{A}}}-\alpha+1}}]$.

We prove that every two ϕ_i and ϕ_j are disjoint for $i \neq j$. Assume that a certain submatrix, say, $\mathbf{A}_j^i$, belongs to both ϕ_i and ϕ_j; then there exist matrices $\mathbf{X}$ and $\mathbf{Y}$ such that

$$\mathbf{A}_j^i = [\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha} \bar{\mathbf{A}}_i] \cdot \mathbf{X} = [\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha} \bar{\mathbf{A}}_j] \cdot \mathbf{Y}.$$

This, in turn, implies that there exists a matrix $\mathbf{Z}$ such that

$$[\bar{\mathbf{A}}_1 \ \ldots \ \bar{\mathbf{A}}_{k'_{\bar{\mathbf{A}}}-\alpha} \ \bar{\mathbf{A}}_i \ \bar{\mathbf{A}}_j] \cdot \mathbf{Z} = \mathbf{0}.$$

This is in contradiction with the definition of $k'_{\bar{\mathbf{A}}}$ and the fact that $\alpha \geqslant 2$.

Let us now count the number of submatrices of $\mathbf{A}$ in the above disjoint sets. In $\{\mathbf{A}_1, \ldots, \mathbf{A}_{k'_{\bar{\mathbf{A}}}-\alpha-\beta}\}$, there are $k'_{\bar{\mathbf{A}}} - \alpha - \beta$ submatrices. In each set ϕ_i there are at least $\beta + 1$ submatrices, and we have $R - k'_{\bar{\mathbf{A}}} + \alpha$ such ϕ_i. Therefore, the total number of submatrices of $\mathbf{A}$ from all disjoint sets is at least

$$k'_{\bar{\mathbf{A}}} - \alpha - \beta + (\beta + 1)(R - k'_{\bar{\mathbf{A}}} + \alpha) = \beta(R - k'_{\bar{\mathbf{A}}}) + R + (\alpha - 1)\beta,$$

which is strictly greater than R for $\alpha \geqslant 2$ and $\beta \geqslant 1$. Obviously, $\mathbf{A}$ has only R submatrices, so we have a contradiction. ∎

3. Rank and k'-rank of Khatri–Rao products of partitioned matrices. In our analysis of the uniqueness of block decompositions [3], we make use of additional lemmas, besides the equivalence lemma for partitioned matrices, that establish certain Khatri–Rao products of partitioned matrices are full column rank. These are derived in the present section.

We start from a lemma that gives a lower-bound on the k-rank of a columnwise Khatri–Rao product. This lemma is proved in [8]. A shorter proof is given in [9, 10]. We give yet another proof, which is easier to generalize to Khatri–Rao products of arbitrarily partitioned matrices.

LEMMA 3.1. *Consider matrices* $\mathbf{A} \in \mathbb{K}^{I \times R}$ *and* $\mathbf{B} \in \mathbb{K}^{J \times R}$.

(i) *If* $k_{\mathbf{A}} = 0$ *or* $k_{\mathbf{B}} = 0$, *then* $k_{\mathbf{A} \odot_c \mathbf{B}} = 0$.

(ii) *If* $k_{\mathbf{A}} \geqslant 1$ *and* $k_{\mathbf{B}} \geqslant 1$, *then* $k_{\mathbf{A} \odot_c \mathbf{B}} \geqslant \min(k_{\mathbf{A}} + k_{\mathbf{B}} - 1, R)$.

Proof. First, we prove (i). If $k_{\mathbf{A}} = 0$, then $\mathbf{A}$ has an all-zero column. Consequently, $\mathbf{A} \odot_c \mathbf{B}$ also has an all-zero column and $k_{\mathbf{A} \odot_c \mathbf{B}} = 0$. The same holds if $k_{\mathbf{B}} = 0$. This completes the proof of (i).

Next, we prove (ii). Suppose $k_{\mathbf{A}} \geqslant 1$ and $k_{\mathbf{B}} \geqslant 1$. Let $m = \min(k_{\mathbf{A}} + k_{\mathbf{B}} - 1, R)$. We have to prove that any set of m columns of $\mathbf{A} \odot_c \mathbf{B}$ is linearly independent. Without loss of generality we prove that this is the case for the first m columns of $\mathbf{A} \odot_c \mathbf{B}$. (Another set of m columns can first be permuted to the first positions. This does not change the k-rank. We can then continue as below.) Let $\mathbf{A}_f = [\mathbf{a}_1 \dots \mathbf{a}_m]$, $\mathbf{B}_f = [\mathbf{b}_1 \dots \mathbf{b}_m]$, $\mathbf{A}_g = [\mathbf{a}_1 \dots \mathbf{a}_{k_{\mathbf{A}}}]$, $\mathbf{B}_g = [\mathbf{b}_{m-k_{\mathbf{B}}+1} \dots \mathbf{b}_m]$. Suppose $\mathbf{U} = (\mathbf{S}\mathbf{A}_f) \odot_c (\mathbf{T}\mathbf{B}_f) = (\mathbf{S} \otimes \mathbf{T})(\mathbf{A}_f \odot_c \mathbf{B}_f)$, where $\mathbf{S} \otimes \mathbf{T}$ is nonsingular if both $\mathbf{S}$ and $\mathbf{T}$ are nonsingular. Premultiplying a matrix by a nonsingular matrix does not change its rank nor its k-rank. Hence the rank of $\mathbf{U}$ is equal to the rank of $\mathbf{A}_f \odot_c \mathbf{B}_f$ if $\mathbf{S}$ and $\mathbf{T}$ are nonsingular. The same holds for the k-rank. We choose $\mathbf{S}$ and $\mathbf{T}$ in the following way:

$$\mathbf{S} = \begin{pmatrix} \mathbf{A}_g^{\dagger} \\ \mathbf{A}_g^{\dagger,\perp} \end{pmatrix} \quad \mathbf{T} = \begin{pmatrix} \mathbf{B}_g^{\dagger} \\ \mathbf{B}_g^{\dagger,\perp} \end{pmatrix} \tag{3.1}$$

in which $\mathbf{A}_g^{\dagger,\perp}$ is an (arbitrary) $((I - k_{\mathbf{A}}) \times I)$ matrix such that $\operatorname{span}[(\mathbf{A}_g^{\dagger,\perp})^T] = \operatorname{null}(\mathbf{A}_g)$, and in which $\mathbf{B}_g^{\dagger,\perp}$ is an (arbitrary) $((J - k_{\mathbf{B}}) \times J)$ matrix such that $\operatorname{span}[(\mathbf{B}_g^{\dagger,\perp})^T] = \operatorname{null}(\mathbf{B}_g)$. If we choose $\mathbf{S}$ and $\mathbf{T}$ this way, $\mathbf{U}$ has a very special structure.

Let us first illustrate this with an example. Assume a matrix $\mathbf{A} \in \mathbb{K}^{2 \times 4}$ with $k_{\mathbf{A}} = 2$ and a matrix $\mathbf{B} \in \mathbb{K}^{3 \times 4}$ with $k_{\mathbf{B}} = 3$. Then we have $\mathbf{A}_f = \mathbf{A}$, $\mathbf{B}_f = \mathbf{B}$, $k_{\mathbf{A}_f} = k_{\mathbf{A}}$ and $k_{\mathbf{B}_f} = k_{\mathbf{B}}$. We now have

$$\tilde{\mathbf{A}} = \mathbf{S} \cdot \mathbf{A}_f = \begin{pmatrix} 1 & 0 & \tilde{a}_{13} & \tilde{a}_{14} \\ 0 & 1 & \tilde{a}_{23} & \tilde{a}_{24} \end{pmatrix},$$

$$\tilde{\mathbf{B}} = \mathbf{T} \cdot \mathbf{B}_f = \begin{pmatrix} \tilde{b}_{11} & 1 & 0 & 0 \\ \tilde{b}_{21} & 0 & 1 & 0 \\ \tilde{b}_{31} & 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{U} = \tilde{\mathbf{A}} \odot_c \tilde{\mathbf{B}} = \begin{pmatrix} \tilde{b}_{11} & 0 & 0 & 0 \\ \tilde{b}_{21} & 0 & \tilde{a}_{13} & 0 \\ \tilde{b}_{31} & 0 & 0 & \tilde{a}_{14} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \tilde{a}_{23} & 0 \\ 0 & 0 & 0 & \tilde{a}_{24} \end{pmatrix}.$$

Note that neither $\tilde{a}_{23}$ nor $\tilde{a}_{24}$ can be equal to zero, otherwise $k_{\tilde{\mathbf{A}}} < 2 = k_{\mathbf{A}_f}$ while $\mathbf{S}$ is nonsingular. On the other hand, $[\tilde{b}_{11}\ \tilde{b}_{21}\ \tilde{b}_{31}]$ cannot be equal to $[0\ 0\ 0]$, otherwise $k_{\tilde{\mathbf{B}}} = 0 < 3 = k_{\mathbf{B}_f}$ while $\mathbf{T}$ is nonsingular. We conclude that $\mathbf{U}$ is full column rank. Since $\mathbf{S}$ and $\mathbf{T}$ are nonsingular, $\mathbf{A}_f \odot_c \mathbf{B}_f$ is also full column rank.

In general, we have

$$\tilde{\mathbf{A}} = \mathbf{S} \cdot \mathbf{A}_f = \begin{pmatrix} \overbrace{\mathbf{I}_{k_\mathbf{A} \times k_\mathbf{A}}}^{k_\mathbf{A}} & \overbrace{\tilde{\mathbf{A}}(1 : k_\mathbf{A}, k_\mathbf{A} + 1 : m)}^{m - k_\mathbf{A}} \\ \mathbf{0}_{(I - k_\mathbf{A}) \times k_\mathbf{A}} & \tilde{\mathbf{A}}(1 + k_\mathbf{A} : I, k_\mathbf{A} + 1 : m) \end{pmatrix},$$

$$\tilde{\mathbf{B}} = \mathbf{T} \cdot \mathbf{B}_f = \begin{pmatrix} \underbrace{\tilde{\mathbf{B}}(1 : k_\mathbf{B}, 1 : m - k_\mathbf{B})}_{m - k_\mathbf{B}} & \underbrace{\mathbf{I}_{k_\mathbf{B} \times k_\mathbf{B}}}_{k_\mathbf{B}} \\ \tilde{\mathbf{B}}(k_\mathbf{B} + 1 : J, 1 : m - k_\mathbf{B}) & \mathbf{0}_{(J - k_\mathbf{B}) \times k_\mathbf{B}} \end{pmatrix}.$$

Key to understanding the structure of $\mathbf{U} = \tilde{\mathbf{A}} \odot_c \tilde{\mathbf{B}}$ is the specific form of the first $k_\mathbf{A}$ columns of $\tilde{\mathbf{A}}$ and the last $k_\mathbf{B}$ columns of $\tilde{\mathbf{B}}$, together with the fact that by definition of m, $m - k_\mathbf{B} < k_\mathbf{A}$ and $m - k_\mathbf{A} < k_\mathbf{B}$. This structure neatly generalizes the structure in the example above. The first $m - k_\mathbf{B}$ columns of $\mathbf{U}$ form a block-diagonal matrix, containing the first $m - k_\mathbf{B}$ columns of $\tilde{\mathbf{B}}$ in the diagonal blocks and zeros below. Each of the next $R - 2m + k_\mathbf{A} + k_\mathbf{B}$ columns of $\mathbf{U}$ is all-zero, except for a single 1 that is also the only nonzero entry of its row. The last $m - k_\mathbf{A}$ columns of $\mathbf{U}$ contain the corresponding entries of $\tilde{\mathbf{A}}(k_\mathbf{A} : I, k_\mathbf{A} + 1 : m)$ in rows where they form the only nonzero entries. The columns of $\tilde{\mathbf{A}}(k_\mathbf{A} : I, k_\mathbf{A} + 1 : m)$ cannot be all-zero. Suppose by contradiction that the nth column of $\tilde{\mathbf{A}}(k_\mathbf{A} : I, k_\mathbf{A} + 1 : m)$ is all-zero. Then the first $k_\mathbf{A} - 1$ columns of $\tilde{\mathbf{A}}$, together with its $(k_\mathbf{A} + n)$th column, form a linearly dependent set. Hence, $k_{\tilde{\mathbf{A}}} < k_\mathbf{A} \leqslant k_{\mathbf{A}_f}$ while $\mathbf{S}$ is nonsingular. We have a contradiction. On the other hand, none of the first $m - k_\mathbf{B}$ columns of $\tilde{\mathbf{B}}$ can be all-zero either, otherwise $k_{\tilde{\mathbf{B}}} = 0 < k_\mathbf{B} \leqslant k_{\mathbf{B}_f}$ while $\mathbf{T}$ is nonsingular. We conclude that $\mathbf{U}$ is full column rank. Hence, $\mathbf{A}_f \odot_c \mathbf{B}_f$ is also full column rank. This completes the proof. ∎

Lemma 3.1 can be generalized to Khatri–Rao products of arbitrarily partitioned matrices as follows.

LEMMA 3.2. *Consider partitioned matrices* $\mathbf{A} = [\mathbf{A}_1\ \ldots\ \mathbf{A}_R]$ *with* $\mathbf{A}_r \in \mathbb{K}^{I \times L_r}$, $1 \leqslant r \leqslant R$, *and* $\mathbf{B} = [\mathbf{B}_1\ \ldots\ \mathbf{B}_R]$ *with* $\mathbf{B}_r \in \mathbb{K}^{J \times M_r}$, $1 \leqslant r \leqslant R$.

(i) *If* $k'_\mathbf{A} = 0$ *or* $k'_\mathbf{B} = 0$, *then* $k'_{\mathbf{A} \odot \mathbf{B}} = 0$.

(ii) *If* $k'_\mathbf{A} \geqslant 1$ *and* $k'_\mathbf{B} \geqslant 1$, *then* $k'_{\mathbf{A} \odot \mathbf{B}} \geqslant \min(k'_\mathbf{A} + k'_\mathbf{B} - 1, R)$.

Proof. We work in analogy with the proof of Lemma 3.1.

First, we prove (i). If $k'_\mathbf{A} = 0$, then $\mathbf{A}$ has a rank-deficient submatrix. Consequently, $\mathbf{A} \odot \mathbf{B}$ also has a rank-deficient submatrix and $k'_{\mathbf{A} \odot \mathbf{B}} = 0$. The same holds if $k'_\mathbf{B} = 0$. This completes the proof of (i).

Next, we prove (ii). Suppose $k'_\mathbf{A} \geqslant 1$ and $k'_\mathbf{B} \geqslant 1$. Let $m = \min(k'_\mathbf{A} + k'_\mathbf{B} - 1, R)$. We have to prove that any set of m submatrices of $\mathbf{A} \odot \mathbf{B}$ yields a linearly independent set of columns. Without loss of generality we prove that this is the case for the first m submatrices of $\mathbf{A} \odot \mathbf{B}$. Let $\mathbf{A}_f = [\mathbf{A}_1 \ldots \mathbf{A}_m]$, $\mathbf{B}_f = [\mathbf{B}_1 \ldots \mathbf{B}_m]$, $\mathbf{A}_g = [\mathbf{A}_1 \ldots \mathbf{A}_{k'_\mathbf{A}}]$, $\mathbf{B}_g = [\mathbf{B}_{m - k'_\mathbf{B} + 1} \ldots \mathbf{B}_m]$. Suppose $\mathbf{U} = (\mathbf{S}\mathbf{A}_f) \odot (\mathbf{T}\mathbf{B}_f) = (\mathbf{S} \otimes \mathbf{T})(\mathbf{A}_f \odot \mathbf{B}_f)$. Hence the rank of $\mathbf{U}$ is equal to the rank of $\mathbf{A}_f \odot \mathbf{B}_f$ if $\mathbf{S}$ and $\mathbf{T}$ are nonsingular. The same holds for the k'-rank. We choose $\mathbf{S}$ and $\mathbf{T}$ as in (3.1). Let $\tilde{\mathbf{A}} = \mathbf{S} \cdot \mathbf{A}_f$ and $\tilde{\mathbf{B}} = \mathbf{T} \cdot \mathbf{B}_f$. The structure of $\mathbf{U}$ allows for a similar reasoning as in Lemma 3.1.

Let us first illustrate this with an example. Assume a matrix $\mathbf{A} \in \mathbb{K}^{4\times 6}$, consisting of 3 (4×2) submatrices, with $k'_{\mathbf{A}} = 2$, and a matrix $\mathbf{B} \in \mathbb{K}^{4\times 6}$, also consisting of three (4×2) submatrices, with $k'_{\mathbf{B}} = 2$. Then we have $\mathbf{A}_f = \mathbf{A}$, $\mathbf{B}_f = \mathbf{B}$, $k'_{\mathbf{A}_f} = k'_{\mathbf{A}}$, and $k'_{\mathbf{B}_f} = k'_{\mathbf{B}}$. We now have

$$\tilde{\mathbf{A}} = \mathbf{S} \cdot \mathbf{A}_f = \left(\begin{array}{cc|cc|cc} 1 & & & & \tilde{a}_{15} & \tilde{a}_{16} \\ & 1 & & & \tilde{a}_{25} & \tilde{a}_{26} \\ \hline & & 1 & & \tilde{a}_{35} & \tilde{a}_{36} \\ & & & 1 & \tilde{a}_{45} & \tilde{a}_{46} \end{array}\right),$$

$$\tilde{\mathbf{B}} = \mathbf{T} \cdot \mathbf{B}_f = \left(\begin{array}{cc|cc|cc} \tilde{b}_{11} & \tilde{b}_{12} & 1 & & & \\ \tilde{b}_{21} & \tilde{b}_{21} & & 1 & & \\ \tilde{b}_{31} & \tilde{b}_{31} & & & 1 & \\ \tilde{b}_{41} & \tilde{b}_{41} & & & & 1 \end{array}\right),$$

$$\mathbf{U} = \tilde{\mathbf{A}} \odot \tilde{\mathbf{B}} = \left(\begin{array}{cc|cc|cc|cccc} \tilde{b}_{11} & \tilde{b}_{12} & & & & & \tilde{a}_{15} & & \tilde{a}_{16} & \\ \tilde{b}_{21} & \tilde{b}_{22} & & & & & & \tilde{a}_{15} & & \tilde{a}_{16} \\ \tilde{b}_{31} & \tilde{b}_{32} & & & & & \tilde{a}_{25} & & \tilde{a}_{26} & \\ \tilde{b}_{41} & \tilde{b}_{42} & & & & & & \tilde{a}_{25} & & \tilde{a}_{26} \\ \hline & & \tilde{b}_{11} & \tilde{b}_{12} & & & \tilde{a}_{35} & & \tilde{a}_{36} & \\ & & \tilde{b}_{21} & \tilde{b}_{22} & & & & \tilde{a}_{35} & & \tilde{a}_{36} \\ & & \tilde{b}_{31} & \tilde{b}_{32} & & & \tilde{a}_{45} & & \tilde{a}_{46} & \\ & & \tilde{b}_{41} & \tilde{b}_{42} & & & & \tilde{a}_{45} & & \tilde{a}_{46} \\ \hline & & & & 1 & & & & & \\ & & & & & 1 & & & & \\ & & & & & & 1 & & & \\ & & & & & & & 1 & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \end{array}\right).$$

Note that $\tilde{\mathbf{A}}(3:4, 5:6)$ cannot be rank-deficient, otherwise $k'_{\tilde{\mathbf{A}}} < 2 = k'_{\mathbf{A}_f}$ while $\mathbf{S}$ is nonsingular. On the other hand, $\tilde{\mathbf{B}}(:, 1:2)$ cannot be rank-deficient, otherwise $k'_{\tilde{\mathbf{B}}} = 0 < 2 = k'_{\mathbf{B}_f}$ while $\mathbf{T}$ is nonsingular. We conclude that $\mathbf{U}$ is full column rank.

In general, the structure of $\mathbf{U}$ is as follows. Its leftmost $m - k'_{\mathbf{B}}$ submatrices form a block-diagonal matrix. The matrices in the diagonal blocks can be rank-deficient only if the corresponding submatrix of $\tilde{\mathbf{B}}$ is rank-deficient. This would imply that $k'_{\tilde{\mathbf{B}}} = 0 < k'_{\mathbf{B}_f}$ while $\mathbf{T}$ is nonsingular. Each column of the next $R - 2m + k'_{\mathbf{A}} + k'_{\mathbf{B}}$ submatrices of $\mathbf{U}$ is all-zero except for a single 1 that is also the only nonzero entry of its row. Consider the partitioning $\tilde{\mathbf{A}}(\sum_{r=1}^{k'_{\mathbf{A}}-1} L_r + 1 : I, \sum_{r=1}^{k'_{\mathbf{A}}} L_r + 1 : \sum_{r=1}^{m} L_r) = [\bar{\mathbf{A}}_{k'_{\mathbf{A}}+1} \dots \bar{\mathbf{A}}_m]$. The matrices $\bar{\mathbf{A}}_{k'_{\mathbf{A}}+1}, \dots, \bar{\mathbf{A}}_m$ can be rank-deficient only if $k'_{\tilde{\mathbf{A}}} < k'_{\mathbf{A}_f}$ while $\mathbf{S}$ is nonsingular. These matrices yield additional independent columns in $\mathbf{U}$. We conclude that $\mathbf{U}$ is full column rank. Hence, $\mathbf{A}_f \odot \mathbf{B}_f$ is full column rank. This completes the proof. □

Lemma 3.2 is a first tool that will be used in [3] to make sure that certain Khatri–Rao products of partitioned matrices are full column rank. Next, we generalize Lemma 2.2 in [2], saying that a columnwise Khatri–Rao product is generically full column rank, to Khatri–Rao products of arbitrarily partitioned matrices.

LEMMA 3.3. *Consider partitioned matrices* $\mathbf{A} = [\mathbf{A}_1 \ \ldots \ \mathbf{A}_R]$ *with* $\mathbf{A}_r \in \mathbb{K}^{I \times L_r}$, $1 \leqslant r \leqslant R$, *and* $\mathbf{B} = [\mathbf{B}_1 \ \ldots \ \mathbf{B}_R]$ *with* $\mathbf{B}_r \in \mathbb{K}^{J \times M_r}$, $1 \leqslant r \leqslant R$. *Generically we have that* $\operatorname{rank}(\mathbf{A} \odot \mathbf{B}) = \min(IJ, \sum_{r=1}^{R} L_r M_r)$.

Proof. We prove the theorem by induction on R.

For $R = 1$, $\mathbf{A}_1$ and $\mathbf{B}_1$ are generically nonsingular. Hence, $\mathbf{A} \odot \mathbf{B} = \mathbf{A}_1 \otimes \mathbf{B}_1$ is generically nonsingular.

Now assume that the lemma holds for $R = 1, 2, \ldots, \tilde{R} - 1$. Then we prove that it also holds for $R = \tilde{R}$. Assume that $IJ \geqslant \sum_{r=1}^{\tilde{R}} L_r M_r$. A similar reasoning applies when $IJ > \sum_{r=1}^{\tilde{R}-1} L_r M_r$ but $IJ < \sum_{r=1}^{\tilde{R}} L_r M_r$. Let the columns of $\mathbf{A}_{\tilde{R}}^{\perp}$ form a basis for $\operatorname{null}(\mathbf{A}_{\tilde{R}})$ and let the columns of $\mathbf{B}_{\tilde{R}}^{\perp}$ form a basis for $\operatorname{null}(\mathbf{B}_{\tilde{R}})$. Define $\tilde{\mathbf{A}} = [\mathbf{A}_{\tilde{R}} \ \mathbf{A}_{\tilde{R}}^{\perp}]$ and $\tilde{\mathbf{B}} = [\mathbf{B}_{\tilde{R}} \ \mathbf{B}_{\tilde{R}}^{\perp}]$. Generically, $\mathbf{A}_{\tilde{R}}$ and $\mathbf{B}_{\tilde{R}}$ are full column rank. Hence, $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{A}} \otimes \tilde{\mathbf{B}}$ are also generically full column rank. Now replace the columns of $\mathbf{A}_{\tilde{R}} \otimes \mathbf{B}_{\tilde{R}}$ in $\tilde{\mathbf{A}} \otimes \tilde{\mathbf{B}}$ by random vectors $\mathbf{v}_j \in \mathbb{K}^{IJ}$, $j = 1, \ldots, L_{\tilde{R}} M_{\tilde{R}}$. Call the resulting matrix $\mathbf{C}$ and define $\mathbf{V} = [\mathbf{v}_1 \ \ldots \ \mathbf{v}_{L_{\tilde{R}} M_{\tilde{R}}}]$. For $\mathbf{C}$ to be rank deficient, a nontrivial linear combination of the columns of $[\mathbf{A}_{\tilde{R}}^{\perp} \otimes \mathbf{B}_{\tilde{R}} \ \ \mathbf{A}_{\tilde{R}} \otimes \mathbf{B}_{\tilde{R}}^{\perp} \ \ \mathbf{A}_{\tilde{R}}^{\perp} \otimes \mathbf{B}_{\tilde{R}}^{\perp}]$ must be in $\operatorname{span}(\mathbf{V})$. This is a probability-zero event. Turned the other way around, if $\mathbf{v}_j \in \mathbb{K}^{IJ}$, $j = 1, \ldots, L_{\tilde{R}} M_{\tilde{R}}$ are a given linearly independent set of vectors and if we randomly choose $\mathbf{A}_{\tilde{R}} \in \mathbb{K}^{I \times L_{\tilde{R}}}$ and $\mathbf{B}_{\tilde{R}} \in \mathbb{K}^{J \times M_{\tilde{R}}}$, then the associated matrix $\mathbf{C}$ is full rank with probability one. Now let the vectors $\mathbf{v}_j$ be orthogonal to $\operatorname{span}(\mathbf{A}_1 \otimes \mathbf{B}_1 \ \ldots \ \mathbf{A}_{\tilde{R}-1} \otimes \mathbf{B}_{\tilde{R}-1})$. Since the intersection of $\operatorname{span}(\mathbf{V})$ and the orthogonal complement of $\mathbf{A}_{\tilde{R}} \otimes \mathbf{B}_{\tilde{R}}$ is generically zero, $\mathbf{V}^T(\mathbf{A}_{\tilde{R}} \otimes \mathbf{B}_{\tilde{R}})$ is generically full rank. In other words, $\mathbf{A}_{\tilde{R}} \otimes \mathbf{B}_{\tilde{R}}$ adds $L_{\tilde{R}} M_{\tilde{R}}$ independent directions to $[\mathbf{A}_1 \otimes \mathbf{B}_1 \ \ldots \ \mathbf{A}_{\tilde{R}-1} \otimes \mathbf{B}_{\tilde{R}-1}]$. Hence, $[\mathbf{A}_1 \otimes \mathbf{B}_1 \ \ldots \ \mathbf{A}_{\tilde{R}} \otimes \mathbf{B}_{\tilde{R}}]$ is generically full column rank. ∎

Acknowledgments. The author wishes to thank A. Stegeman (Heijmans Institute, The Netherlands) for proofreading an early version of the manuscript. A large part of this research was carried out when L. De Lathauwer was with the French Centre National de la Recherche Scientifique (C.N.R.S.).

REFERENCES

[1] J. CARROLL AND J. CHANG, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 9 (1970), pp. 267–283.

[2] L. DE LATHAUWER, *A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666.

[3] L. DE LATHAUWER, *Decompositions of a higher-order tensor in block terms—Part* II: *Definitions and uniqueness*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1033–1066.

[4] T. JIANG AND N.D. SIDIROPOULOS, *Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear models with constant modulus constraints*, IEEE Trans. Signal Process., 52 (2004), pp. 2625–2636.

[5] R.A. HARSHMAN, *Foundations of the PARAFAC procedure*: *Model and conditions for an "explanatory" multi-mode factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[6] J.B. KRUSKAL, *Three-way arrays*: *rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.

[7] C.R. RAO AND S.K. MITRA, *Generalized Inverse of Matrices and Its Applications*, John Wiley and Sons, New York, 1971.

[8] N.D. SIDIROPOULOS AND R. BRO, *On the uniqueness of multilinear decomposition of N-way arrays*, J. Chemometrics, 14 (2000), pp. 229–239.

[9] A. STEGEMAN AND N.D. SIDIROPOULOS, *On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition*, Linear Algebra Appl., 420 (2007), pp. 540–552.

[10] J.M.F. Ten Berge, *The K-Rank of a Khatri–Rao Product*, Tech. report, Heijmans Institute of Psychological Research, University of Groningen, Groningen, the Netherlands, 2000.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1033–1066

DECOMPOSITIONS OF A HIGHER-ORDER TENSOR IN BLOCK TERMS—PART II: DEFINITIONS AND UNIQUENESS*

LIEVEN DE LATHAUWER†

Abstract. In this paper we introduce a new class of tensor decompositions. Intuitively, we decompose a given tensor block into blocks of smaller size, where the size is characterized by a set of mode-n ranks. We study different types of such decompositions. For each type we derive conditions under which essential uniqueness is guaranteed. The parallel factor decomposition and Tucker's decomposition can be considered as special cases in the new framework. The paper sheds new light on fundamental aspects of tensor algebra.

Key words. multilinear algebra, higher-order tensor, Tucker decomposition, canonical decomposition, parallel factors model

AMS subject classifications. 15A18, 15A69

DOI. 10.1137/070690729

1. Introduction.

The two main tensor generalizations of the matrix singular value decomposition (SVD) are, on one hand, the Tucker decomposition/higher-order singular value decomposition (HOSVD) [59, 60, 12, 13, 15] and, on the other hand, the canonical/parallel factor (CANDECOMP/PARAFAC) decomposition [7, 26]. These are connected with two different tensor generalizations of the concept of matrix rank. The Tucker decomposition/HOSVD is linked with the set of mode-n ranks, which generalize column rank, row rank, etc. CANDECOMP/PARAFAC has to do with rank in the meaning of the minimal number of rank-1 terms that are needed in an expansion of the matrix/tensor. In this paper we introduce a new class of tensor SVDs, which we call block term decompositions. These lead to a framework that unifies the Tucker decomposition/HOSVD and CANDECOMP/PARAFAC. Block term decompositions also provide a unifying view on tensor rank.

We study different types of block term decompositions. For each type, we derive sufficient conditions for essential uniqueness, i.e., uniqueness up to trivial indeterminacies. We derive two types of uniqueness conditions. The first type follows from a reasoning that involves invariant subspaces associated with the tensor. This type of conditions generalizes the result on CANDECOMP/PARAFAC uniqueness that is presented in [6, 40, 47, 48]. The second type generalizes Kruskal's condition for CANDECOMP/PARAFAC uniqueness, discussed in [38, 49, 54].

In the following subsection we explain our notation and introduce some basic definitions. In subsection 1.2 we recall the Tucker decomposition/HOSVD and also the CANDECOMP/PARAFAC decomposition and summarize some of their properties.

*Received by the editors May 7, 2007; accepted for publication (in revised form) by J. G. Nagy April 14, 2008; published electronically September 25, 2008. This research was supported by Research Council K.U.Leuven: GOA-Ambiorics, CoE EF/05/006 Optimization in Engineering (OPTEC), CIF1; F.W.O.: project G.0321.06 and Research Communities ICCoS, ANMMM, and MLDM; the Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, "Dynamical systems, control and optimization," 2007–2011); and EU: ERNSI.

http://www.siam.org/journals/simax/30-3/69072.html

†Subfaculty Science and Technology, Katholieke Universiteit Leuven Campus Kortrijk, E. Sabbelaan 53, 8500 Kortrijk, Belgium (Lieven.DeLathauwer@kuleuven-kortrijk.be), and Department of Electrical Engineering (ESAT), Research Division SCD, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (Lieven.DeLathauwer@esat.kuleuven.be, http://homes.esat.kuleuven.be/∼delathau/home.html).

In section 2 we define block term decompositions. We subsequently introduce decomposition in rank-$(L, L, 1)$ terms (subsection 2.1), decomposition in rank-(L, M, N) terms (subsection 2.2), and type-2 decomposition in rank-$(L, M, \cdot)$ terms (subsection 2.3). The uniqueness of these decompositions is studied in sections 4, 5, and 6, respectively. In the analysis we use some tools that have been introduced in [19]. These will briefly be recalled in section 3.

Several proofs of lemmas and theorems establishing Kruskal-type conditions for essential uniqueness of the new decompositions generalize results for PARAFAC presented in [54]. We stay quite close to the text of [54]. We recommend studying the proofs in [54] before reading this paper.

1.1. Notation and basic definitions.

1.1.1. Notation. We use $\mathbb{K}$ to denote $\mathbb{R}$ or $\mathbb{C}$ when the difference is not important. In this paper scalars are denoted by lowercase letters (a, b, ...), vectors are written in boldface lowercase ($\mathbf{a}$, $\mathbf{b}$, ...), matrices correspond to boldface capitals ($\mathbf{A}$, $\mathbf{B}$, ...), and tensors are written as calligraphic letters ($\mathcal{A}$, $\mathcal{B}$, ...). This notation is consistently used for lower-order parts of a given structure. For instance, the entry with row index i and column index j in a matrix $\mathbf{A}$, i.e., $(\mathbf{A})_{ij}$, is symbolized by a_{ij} (also $(\mathbf{a})_i = a_i$ and $(\mathcal{A})_{ijk} = a_{ijk}$). If no confusion is possible, the ith column vector of a matrix $\mathbf{A}$ is denoted as $\mathbf{a}_i$, i.e., $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \ldots]$. Sometimes we will use the MATLAB colon notation to indicate submatrices of a given matrix or subtensors of a given tensor. Italic capitals are also used to denote index upper bounds (e.g., $i = 1, 2, \ldots, I$). The symbol $\otimes$ denotes the Kronecker product,

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \ldots \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \ldots \\ \vdots & \vdots & \end{pmatrix}.$$

Let $\mathbf{A} = [\mathbf{A}_1\ \ldots\ \mathbf{A}_R]$ and $\mathbf{B} = [\mathbf{B}_1\ \ldots\ \mathbf{B}_R]$ be two partitioned matrices. Then the Khatri–Rao product is defined as the partitionwise Kronecker product and represented by $\odot$ [46]:

$$\mathbf{A} \odot \mathbf{B} = (\mathbf{A}_1 \otimes \mathbf{B}_1 \ldots \mathbf{A}_R \otimes \mathbf{B}_R). \tag{1.1}$$

In recent years, the term "Khatri–Rao product" and the symbol $\odot$ have been used mainly in the case where $\mathbf{A}$ and $\mathbf{B}$ are partitioned into vectors. For clarity, we denote this particular, columnwise, Khatri–Rao product by $\odot_c$:

$$\mathbf{A} \odot_c \mathbf{B} = (\mathbf{a}_1 \otimes \mathbf{b}_1 \ldots \mathbf{a}_R \otimes \mathbf{b}_R).$$

The column space of a matrix and its orthogonal complement will be denoted by $\mathrm{span}(\mathbf{A})$ and $\mathrm{null}(\mathbf{A})$. The rank of a matrix $\mathbf{A}$ will be denoted by $\mathrm{rank}(\mathbf{A})$ or $r_{\mathbf{A}}$. The superscripts $\cdot^T$, $\cdot^H$, and $\cdot^\dagger$ denote the transpose, complex conjugated transpose, and Moore–Penrose pseudoinverse, respectively. The operator $\mathrm{diag}(\cdot)$ stacks its scalar arguments in a square diagonal matrix. Analogously, $\mathrm{blockdiag}(\cdot)$ stacks its vector or matrix arguments in a block-diagonal matrix. For vectorization of a matrix $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \ldots]$ we stick to the following convention: $\mathrm{vec}(\mathbf{A}) = [\mathbf{a}_1^T\ \mathbf{a}_2^T\ \ldots]^T$. The symbol δ_{ij} stands for the Kronecker delta, i.e., $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. The $(N \times N)$ identity matrix is represented by $\mathbf{I}_{N\times N}$. The $(I \times J)$ zero matrix is denoted by $\mathbf{0}_{I\times J}$. $\mathbf{1}_N$ is a column vector of all ones of length N. The zero tensor is denoted by $\mathcal{O}$.

1.1.2. Basic definitions.

DEFINITION 1.1. *Consider* $\mathcal{T} \in \mathbb{K}^{I_1 \times I_2 \times I_3}$ *and* $\mathbf{A} \in \mathbb{K}^{J_1 \times I_1}$, $\mathbf{B} \in \mathbb{K}^{J_2 \times I_2}$, $\mathbf{C} \in \mathbb{K}^{J_3 \times I_3}$. *Then the* Tucker mode-1 product $\mathcal{T} \bullet_1 \mathbf{A}$, mode-2 product $\mathcal{T} \bullet_2 \mathbf{B}$, and mode-3 product $\mathcal{T} \bullet_3 \mathbf{C}$ *are defined by*

$$(\mathcal{T} \bullet_1 \mathbf{A})_{j_1 i_2 i_3} = \sum_{i_1=1}^{I_1} t_{i_1 i_2 i_3} a_{j_1 i_1} \qquad \forall j_1, i_2, i_3,$$

$$(\mathcal{T} \bullet_2 \mathbf{B})_{i_1 j_2 i_3} = \sum_{i_2=1}^{I_2} t_{i_1 i_2 i_3} b_{j_2 i_2} \qquad \forall i_1, j_2, i_3,$$

$$(\mathcal{T} \bullet_3 \mathbf{C})_{i_1 i_2 j_3} = \sum_{i_3=1}^{I_3} t_{i_1 i_2 i_3} c_{j_3 i_3} \qquad \forall i_1, i_2, j_3,$$

respectively [11].

In this paper we denote the Tucker mode-n product in the same way as in [10]; in the literature the symbol $\times_n$ is sometimes used [12, 13, 15].

DEFINITION 1.2. *The* Frobenius norm *of a tensor* $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ *is defined as*

$$\|\mathcal{T}\| = \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} |t_{ijk}|^2 \right)^{\frac{1}{2}}.$$

DEFINITION 1.3. *The outer product* $\mathcal{A} \circ \mathcal{B}$ *of a tensor* $\mathcal{A} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_P}$ *and a tensor* $\mathcal{B} \in \mathbb{K}^{J_1 \times J_2 \times \cdots \times J_Q}$ *is the tensor defined by*

$$(\mathcal{A} \circ \mathcal{B})_{i_1 i_2 \ldots i_P j_1 j_2 \ldots j_Q} = a_{i_1 i_2 \ldots i_P} b_{j_1 j_2 \ldots j_Q}$$

for all values of the indices.

For instance, the outer product $\mathcal{T}$ of three vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ is defined by $t_{ijk} = a_i b_j c_k$ for all values of the indices.

DEFINITION 1.4. *A* mode-n vector *of a tensor* $\mathcal{T} \in \mathbb{K}^{I_1 \times I_2 \times I_3}$ *is an* I_n*-dimensional vector obtained from* $\mathcal{T}$ *by varying the index* i_n *and keeping the other indices fixed* [34].

Mode-n vectors generalize column and row vectors.

DEFINITION 1.5. *The* mode-n rank *of a tensor* $\mathcal{T}$ *is the dimension of the subspace spanned by its mode-*n *vectors.*

The mode-n rank of a higher-order tensor is the obvious generalization of the column (row) rank of a matrix.

DEFINITION 1.6. *A third-order tensor is* rank-(L, M, N) *if its mode-*1 *rank, mode-*2 *rank, and mode-*3 *rank are equal to* L, M, *and* N, *respectively.*

A rank-$(1, 1, 1)$ tensor is briefly called rank-1. This definition is equivalent to the following.

DEFINITION 1.7. *A third-order tensor* $\mathcal{T}$ *has* rank 1 *if it equals the outer product of* 3 *vectors.*

The rank (as opposed to mode-n rank) is now defined as follows.

DEFINITION 1.8. *The* rank *of a tensor* $\mathcal{T}$ *is the minimal number of rank-*1 *tensors that yield* $\mathcal{T}$ *in a linear combination* [38].

The following definition has proved useful in the analysis of PARAFAC uniqueness [38, 49, 51, 54].

DEFINITION 1.9. *The* Kruskal rank *or* k-rank *of a matrix* $\mathbf{A}$*, denoted by* $\text{rank}_k(\mathbf{A})$ *or* $k_{\mathbf{A}}$*, is the maximal number* r *such that any set of* r *columns of* $\mathbf{A}$ *is linearly independent* [38].

We call a property generic when it holds with probability one when the parameters of the problem are drawn from continuous probability density functions. Let $\mathbf{A} \in \mathbb{K}^{I\times R}$. Generically, we have $k_{\mathbf{A}} = \min(I, R)$.

It will sometimes be useful to express tensor properties in terms of matrices and vectors. We therefore define standard matrix representations of a third-order tensor.

DEFINITION 1.10. *The standard* $(JK \times I)$ *matrix representation* $(\mathcal{T})_{JK\times I} = \mathbf{T}_{JK\times I}$*,* $(KI \times J)$ *representation* $(\mathcal{T})_{KI\times J} = \mathbf{T}_{KI\times J}$*, and* $(IJ \times K)$ *representation* $(\mathcal{T})_{IJ\times K} = \mathbf{T}_{IJ\times K}$ *of a tensor* $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ *are defined by*

$$\begin{aligned}(\mathbf{T}_{JK\times I})_{(j-1)K+k,i} &= (\mathcal{T})_{ijk},\\ (\mathbf{T}_{KI\times J})_{(k-1)I+i,j} &= (\mathcal{T})_{ijk},\\ (\mathbf{T}_{IJ\times K})_{(i-1)J+j,k} &= (\mathcal{T})_{ijk}\end{aligned}$$

for all values of the indices [34].

Note that in these definitions indices to the right vary more rapidly than indices to the left. Further, the ith $(J \times K)$ matrix slice of $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ will be denoted as $\mathbf{T}_{J\times K,i}$. Similarly, the jth $(K \times I)$ slice and the kth $(I \times J)$ slice will be denoted by $\mathbf{T}_{K\times I,j}$ and $\mathbf{T}_{I\times J,k}$, respectively.

1.2. HOSVD and PARAFAC. We have now enough material to introduce the Tucker/HOSVD [12, 13, 15, 59, 60] and CANDECOMP/PARAFAC [7, 26] decompositions.

DEFINITION 1.11. *A* Tucker decomposition *of a tensor* $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ *is a decomposition of* $\mathcal{T}$ *of the form*

$$\mathcal{T} = \mathcal{D} \bullet_1 \mathbf{A} \bullet_2 \mathbf{B} \bullet_3 \mathbf{C}. \tag{1.2}$$

An HOSVD is a Tucker decomposition, normalized in a particular way. The normalization was suggested in the computational strategy in [59, 60].

DEFINITION 1.12. *An* HOSVD *of a tensor* $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ *is a decomposition of* $\mathcal{T}$ *of the form*

$$\mathcal{T} = \mathcal{D} \bullet_1 \mathbf{A} \bullet_2 \mathbf{B} \bullet_3 \mathbf{C}, \tag{1.3}$$

in which

- *the matrices* $\mathbf{A} \in \mathbb{K}^{I\times L}$*,* $\mathbf{B} \in \mathbb{K}^{J\times M}$*, and* $\mathbf{C} \in \mathbb{K}^{K\times N}$ *are columnwise orthonormal,*
- *the core tensor* $\mathcal{D} \in \mathbb{K}^{L\times M\times N}$ *is*
  - *all-orthogonal,*

$$\begin{aligned}\langle \mathbf{D}_{M\times N,l_1}, \mathbf{D}_{M\times N,l_2}\rangle = \text{trace}(\mathbf{D}_{M\times N,l_1} \cdot \mathbf{D}^H_{M\times N,l_2}) = \sigma^{(1)^2}_{l_1}\delta_{l_1,l_2},\\ 1 \leqslant l_1, l_2 \leqslant L,\\ \langle \mathbf{D}_{N\times L,m_1}, \mathbf{D}_{N\times L,m_2}\rangle = \text{trace}(\mathbf{D}_{N\times L,m_1} \cdot \mathbf{D}^H_{N\times L,m_2}) = \sigma^{(2)^2}_{m_1}\delta_{m_1,m_2},\\ 1 \leqslant m_1, m_2 \leqslant M,\\ \langle \mathbf{D}_{I\times J,n_1}, \mathbf{D}_{I\times J,n_2}\rangle = \text{trace}(\mathbf{D}_{L\times M,n_1} \cdot \mathbf{D}^H_{L\times M,n_2}) = \sigma^{(3)^2}_{n_1}\delta_{n_1,n_2},\\ 1 \leqslant n_1, n_2 \leqslant N,\end{aligned}$$

– *ordered*,

$$\sigma_1^{(1)^2} \geqslant \sigma_2^{(1)^2} \geqslant \ldots \geqslant \sigma_L^{(1)^2} \geqslant 0,$$
$$\sigma_1^{(2)^2} \geqslant \sigma_2^{(2)^2} \geqslant \ldots \geqslant \sigma_M^{(2)^2} \geqslant 0,$$
$$\sigma_1^{(3)^2} \geqslant \sigma_2^{(3)^2} \geqslant \ldots \geqslant \sigma_N^{(3)^2} \geqslant 0.$$

The decomposition is visualized in Figure 1.1.

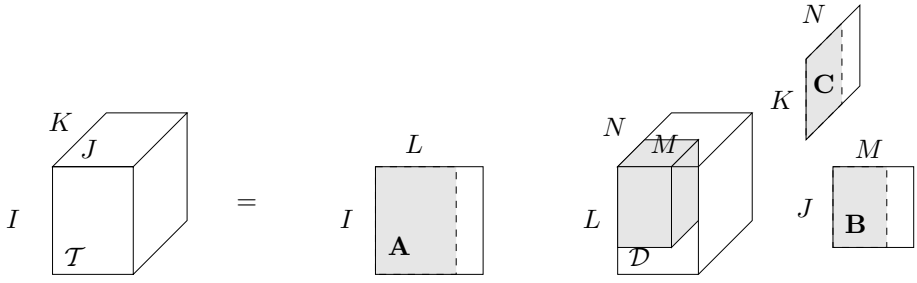


FIG. 1.1. *Visualization of the HOSVD/Tucker decomposition.*

Equation (1.3) can be written in terms of the standard $(JK \times I)$, $(KI \times J)$, and $(IJ \times K)$ matrix representations of $\mathcal{T}$ as follows:

$$\mathbf{T}_{JK\times I} = (\mathbf{B} \otimes \mathbf{C}) \cdot \mathbf{D}_{MN\times L} \cdot \mathbf{A}^T, \tag{1.4}$$

$$\mathbf{T}_{KI\times J} = (\mathbf{C} \otimes \mathbf{A}) \cdot \mathbf{D}_{NL\times M} \cdot \mathbf{B}^T, \tag{1.5}$$

$$\mathbf{T}_{IJ\times K} = (\mathbf{A} \otimes \mathbf{B}) \cdot \mathbf{D}_{LM\times N} \cdot \mathbf{C}^T. \tag{1.6}$$

The HOSVD exists for any $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$. The values L, M, and N correspond to the rank of $\mathbf{T}_{JK\times I}$, $\mathbf{T}_{KI\times J}$, and $\mathbf{T}_{IJ\times K}$, i.e., they are equal to the mode-1, mode-2 and mode-3 rank of $\mathcal{T}$, respectively. In [12] it has been demonstrated that the SVD of matrices and the HOSVD of higher-order tensors have some analogous properties.

Define $\tilde{\mathcal{D}} = \mathcal{D} \bullet_3 \mathbf{C}$. Then

$$\mathcal{T} = \tilde{\mathcal{D}} \bullet_1 \mathbf{A} \bullet_2 \mathbf{B} \tag{1.7}$$

is a (normalized) *Tucker*-2 *decomposition* of $\mathcal{T}$. This decomposition is visualized in Figure 1.2.

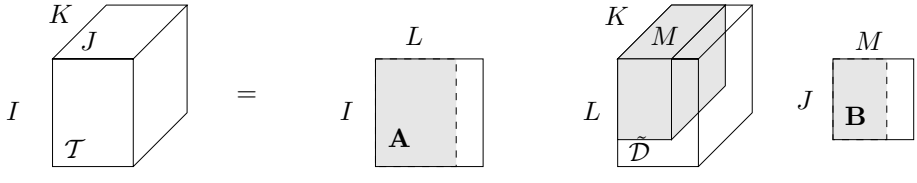


FIG. 1.2. *Visualization of the (normalized) Tucker*-2 *decomposition.*

Besides the HOSVD, there exist other ways to generalize the SVD of matrices. The most well known is CANDECOMP/PARAFAC [7, 26].

DEFINITION 1.13. *A* canonical or parallel factor decomposition (CANDECOMP/PARAFAC) *of a tensor* $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ *is a decomposition of* $\mathcal{T}$ *as a linear combination*

*of rank-*1 *terms*:

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{1.8}$$

The decomposition is visualized in Figure 1.3.

In terms of the standard matrix representations of $\mathcal{T}$, decomposition (1.8) can be written as

$$\mathbf{T}_{JK\times I} = (\mathbf{B} \odot_c \mathbf{C}) \cdot \mathbf{A}^T, \tag{1.9}$$

$$\mathbf{T}_{KI\times J} = (\mathbf{C} \odot_c \mathbf{A}) \cdot \mathbf{B}^T, \tag{1.10}$$

$$\mathbf{T}_{IJ\times K} = (\mathbf{A} \odot_c \mathbf{B}) \cdot \mathbf{C}^T. \tag{1.11}$$

In terms of the $(J \times K)$, $(K \times I)$, and $(I \times J)$ matrix slices of $\mathcal{T}$, we have

$$\mathbf{T}_{J\times K,i} = \mathbf{B} \cdot \mathrm{diag}(a_{i1}, \ldots, a_{iR}) \cdot \mathbf{C}^T, \quad i = 1, \ldots, I. \tag{1.12}$$

$$\mathbf{T}_{K\times I,j} = \mathbf{C} \cdot \mathrm{diag}(b_{j1}, \ldots, b_{jR}) \cdot \mathbf{A}^T, \quad j = 1, \ldots, J. \tag{1.13}$$

$$\mathbf{T}_{I\times J,k} = \mathbf{A} \cdot \mathrm{diag}(c_{k1}, \ldots, c_{kR}) \cdot \mathbf{B}^T, \quad k = 1, \ldots, K. \tag{1.14}$$

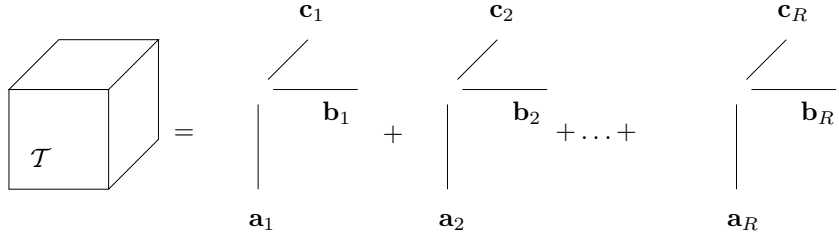


FIG. 1.3. *Visualization of the CANDECOMP/PARAFAC decomposition.*

The fully symmetric variant of PARAFAC, in which $\mathbf{a}_r = \mathbf{b}_r = \mathbf{c}_r$, $r = 1, \ldots, R$, was studied in the nineteenth century in the context of invariant theory [9]. The unsymmetric decomposition was introduced by F. L. Hitchcock in 1927 [27, 28]. Around 1970, the unsymmetric decomposition was independently reintroduced in psychometrics [7] and phonetics [26]. Later, the decomposition was applied in chemometrics and the food industry [1, 5, 53]. In these various disciplines PARAFAC is used for the purpose of multiway factor analysis. The term "canonical decomposition" is standard in psychometrics, while in chemometrics the decomposition is called a parallel factors model. PARAFAC has found important applications in signal processing and data analysis [37]. In wireless telecommunications, it provides powerful means for the exploitation of different types of diversity [49, 50, 18]. It also describes the basic structure of higher-order cumulants of multivariate data on which all algebraic methods for independent component analysis (ICA) are based [8, 14, 29]. Moreover, the decomposition is finding its way to scientific computing, where it leads to a way around the curse of dimensionality [2, 3, 24, 25, 33].

To a large extent, the practical importance of PARAFAC stems from its uniqueness properties. It is clear that one can arbitrarily permute the different rank-1 terms. Also, the factors of a same rank-1 term may be arbitrarily scaled, as long as their product remains the same. We call a PARAFAC decomposition essentially unique when it is subject only to these trivial indeterminacies. The following theorem establishes a condition under which essential uniqueness is guaranteed.

THEOREM 1.14. *The PARAFAC decomposition* (1.8) *is essentially unique if*

$$k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geqslant 2R + 2. \tag{1.15}$$

This theorem was first proved for real tensors in [38]. A concise proof that also applies to complex tensors was given in [49]; in this proof, the permutation lemma of [38] was used. The result was generalized to tensors of arbitrary order in [51]. An alternative proof of the permutation lemma was given in [31]. The overall proof was reformulated in terms of accessible basic linear algebra in [54]. In [17] we derived a more relaxed uniqueness condition that applies when $\mathcal{T}$ is tall in one mode (meaning that, for instance, $K \geqslant R$).

2. Block term decompositions.

2.1. Decomposition in rank-$(L, L, 1)$ terms.

DEFINITION 2.1. *A* decomposition of a tensor $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ in a sum of rank-$(L, L, 1)$ terms *is a decomposition of* $\mathcal{T}$ *of the form*

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{E}_r \circ \mathbf{c}_r, \tag{2.1}$$

in which the $(I \times J)$ *matrices* $\mathbf{E}_r$ *are rank-*L.

We also consider the decomposition of a tensor in a sum of matrix-vector outer products, in which the different matrices do not necessarily all have the same rank.

DEFINITION 2.2. *A* decomposition of a tensor $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ in a sum of rank-$(L_r, L_r, 1)$ terms, $1 \leqslant r \leqslant R$, *is a decomposition of* $\mathcal{T}$ *of the form*

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{E}_r \circ \mathbf{c}_r, \tag{2.2}$$

in which the $(I \times J)$ *matrix* $\mathbf{E}_r$ *is rank-*L_r, $1 \leqslant r \leqslant R$.

If we factorize $\mathbf{E}_r$ as $\mathbf{A}_r \cdot \mathbf{B}_r^T$, in which the matrix $\mathbf{A}_r \in \mathbb{K}^{I \times L_r}$ and the matrix $\mathbf{B}_r \in \mathbb{K}^{J \times L_r}$ are rank-L_r, $r = 1, \ldots, R$, then we can write (2.2) as

$$\mathcal{T} = \sum_{r=1}^{R} (\mathbf{A}_r \cdot \mathbf{B}_r^T) \circ \mathbf{c}_r. \tag{2.3}$$

Define $\mathbf{A} = [\mathbf{A}_1 \ldots \mathbf{A}_R]$, $\mathbf{B} = [\mathbf{B}_1 \ldots \mathbf{B}_R]$, $\mathbf{C} = [\mathbf{c}_1 \ldots \mathbf{c}_R]$. In terms of the standard matrix representations of $\mathcal{T}$, (2.3) can be written as

$$\mathbf{T}_{IJ \times K} = [(\mathbf{A}_1 \odot_c \mathbf{B}_1)\mathbf{1}_{L_1} \ \ldots \ (\mathbf{A}_R \odot_c \mathbf{B}_R)\mathbf{1}_{L_R}] \cdot \mathbf{C}^T, \tag{2.4}$$

$$\mathbf{T}_{JK \times I} = (\mathbf{B} \odot \mathbf{C}) \cdot \mathbf{A}^T, \tag{2.5}$$

$$\mathbf{T}_{KI \times J} = (\mathbf{C} \odot \mathbf{A}) \cdot \mathbf{B}^T. \tag{2.6}$$

In terms of the matrix slices of $\mathcal{T}$, (2.3) can be written as

$$\mathbf{T}_{J \times K, i} = \mathbf{B} \cdot \mathrm{blockdiag}([(\mathbf{A}_1)_{i1} \ \ldots \ (\mathbf{A}_1)_{iL_1}]^T, \ldots, [(\mathbf{A}_R)_{i1} \ \ldots \ (\mathbf{A}_R)_{iL_R}]^T) \cdot \mathbf{C}^T, \quad i = 1, \ldots, I, \tag{2.7}$$

$$\mathbf{T}_{K \times I, j} = \mathbf{C} \cdot \mathrm{blockdiag}([(\mathbf{B}_1)_{j1} \ \ldots \ (\mathbf{B}_1)_{jL_1}], \ldots, [(\mathbf{B}_R)_{j1} \ \ldots \ (\mathbf{B}_R)_{jL_R}]) \cdot \mathbf{A}^T, \quad j = 1, \ldots, J, \tag{2.8}$$

$$\mathbf{T}_{I \times J, k} = \mathbf{A} \cdot \mathrm{blockdiag}(c_{k1}\mathbf{I}_{L_1 \times L_1}, \ldots, c_{kR}\mathbf{I}_{L_R \times L_R}) \cdot \mathbf{B}^T, \quad k = 1, \ldots, K. \tag{2.9}$$

It is clear that in (2.3) one can arbitrarily permute the different rank-$(L_r, L_r, 1)$ terms. Also, one can postmultiply $\mathbf{A}_r$ by any nonsingular $(L_r \times L_r)$ matrix $\mathbf{F}_r \in \mathbb{K}^{L_r \times L_r}$, provided $\mathbf{B}_r$ is premultiplied by the inverse of $\mathbf{F}_r$. Moreover, the factors of a same rank-$(L_r, L_r, 1)$ term may be arbitrarily scaled, as long as their product remains the same. We call the decomposition essentially unique when it is subject only to these trivial indeterminacies. Two representations $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$ that are the same up to trivial indeterminacies are called essentially equal. We (partially) normalize the representation of (2.2) as follows. Scale/counterscale the vectors $\mathbf{c}_r$ and the matrices $\mathbf{E}_r$ such that $\mathbf{c}_r$ are unit-norm. Further, let $\mathbf{E}_r = \mathbf{A}_r \cdot \mathbf{D}_r \cdot \mathbf{B}_r^T$ denote the SVD of $\mathbf{E}_r$. The diagonal matrix $\mathbf{D}_r$ can be interpreted as an $(L_r \times L_r \times 1)$ tensor. Then (2.2) is equivalent to

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r \bullet_3 \mathbf{c}_r. \tag{2.10}$$

Note that in this equation each term is represented in HOSVD form. The decomposition is visualized in Figure 2.1.

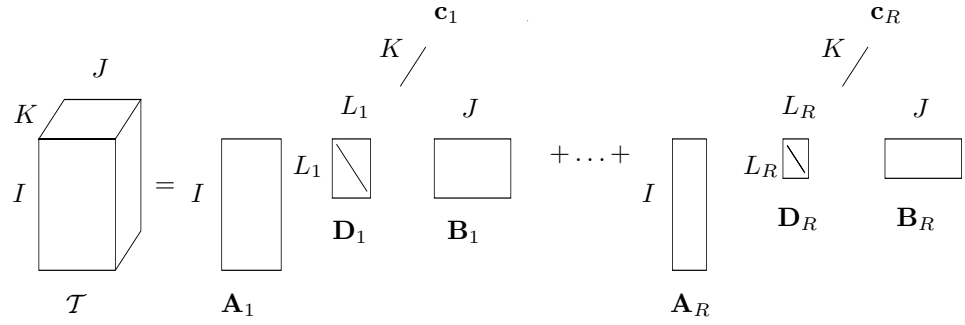


FIG. 2.1. *Visualization of the decomposition of a tensor in a sum of rank-$(L_r, L_r, 1)$ terms, $1 \leqslant r \leqslant R$.*

2.2. Decomposition in rank-(L, M, N) terms.

DEFINITION 2.3. *A* decomposition of a tensor $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ in a sum of rank-(L, M, N) terms *is a decomposition of $\mathcal{T}$ of the form*

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r \bullet_3 \mathbf{C}_r, \tag{2.11}$$

in which $\mathcal{D}_r \in \mathbb{K}^{L \times M \times N}$ are full rank-(L, M, N) and in which $\mathbf{A}_r \in \mathbb{K}^{I \times L}$ (with $I \geqslant L$), $\mathbf{B}_r \in \mathbb{K}^{J \times M}$ (with $J \geqslant M$), and $\mathbf{C}_r \in \mathbb{K}^{K \times N}$ (with $K \geqslant N$) are full column rank, $1 \leqslant r \leqslant R$.

Remark 1. One could also consider a decomposition in rank-(L_r, M_r, N_r) terms, where the different terms possibly have different mode-n ranks. In this paper we focus on the decomposition in rank-(L, M, N) terms.

Define partitioned matrices $\mathbf{A} = [\mathbf{A}_1 \ldots \mathbf{A}_R]$, $\mathbf{B} = [\mathbf{B}_1 \ldots \mathbf{B}_R]$, and $\mathbf{C} = [\mathbf{C}_1 \ldots \mathbf{C}_R]$. In terms of the standard matrix representations of $\mathcal{T}$, (2.11) can be written as

$$\mathbf{T}_{JK \times I} = (\mathbf{B} \odot \mathbf{C}) \cdot \text{blockdiag}((\mathcal{D}_1)_{MN \times L}, \ldots, (\mathcal{D}_R)_{MN \times L}) \cdot \mathbf{A}^T, \tag{2.12}$$

$$\mathbf{T}_{KI \times J} = (\mathbf{C} \odot \mathbf{A}) \cdot \text{blockdiag}((\mathcal{D}_1)_{NL \times M}, \ldots, (\mathcal{D}_R)_{NL \times M}) \cdot \mathbf{B}^T, \tag{2.13}$$

$$\mathbf{T}_{IJ \times K} = (\mathbf{A} \odot \mathbf{B}) \cdot \text{blockdiag}((\mathcal{D}_1)_{LM \times N}, \ldots, (\mathcal{D}_R)_{LM \times N}) \cdot \mathbf{C}^T. \tag{2.14}$$

It is clear that in (2.11) one can arbitrarily permute the different terms. Also, one can postmultiply $\mathbf{A}_r$ by a nonsingular matrix $\mathbf{F}_r \in \mathbb{K}^{L\times L}$, $\mathbf{B}_r$ by a nonsingular matrix $\mathbf{G}_r \in \mathbb{K}^{M\times M}$, and $\mathbf{C}_r$ by a nonsingular matrix $\mathbf{H}_r \in \mathbb{K}^{N\times N}$, provided $\mathcal{D}_r$ is replaced by $\mathcal{D}_r \bullet_1 \mathbf{F}_r^{-1} \bullet_2 \mathbf{G}_r^{-1} \bullet_3 \mathbf{H}_r^{-1}$. We call the decomposition essentially unique when it is subject only to these trivial indeterminacies. We can (partially) normalize (2.11) by representing each term by its HOSVD. The decomposition is visualized in Figure 2.2.

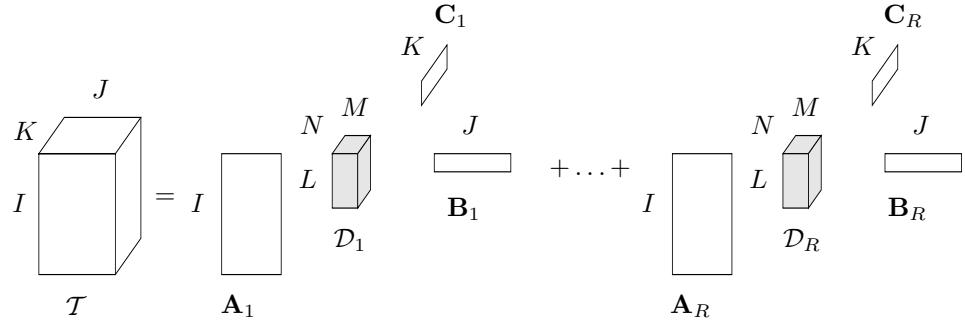


FIG. 2.2. *Visualization of the decomposition of a tensor in a sum of rank-(L, M, N) terms.*

Define $\mathcal{D} = \text{blockdiag}(\mathcal{D}_1, \ldots, \mathcal{D}_R)$. Equation (2.11) can now also be seen as the multiplication of a block-diagonal core tensor $\mathcal{D}$ by means of factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$:

$$\mathcal{T} = \mathcal{D} \bullet_1 \mathbf{A} \bullet_2 \mathbf{B} \bullet_3 \mathbf{C}. \tag{2.15}$$

This alternative interpretation of the decomposition is visualized in Figure 2.3. Two representations $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathcal{D})$ and $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathcal{D}})$ that are the same up to trivial indeterminacies are called essentially equal.

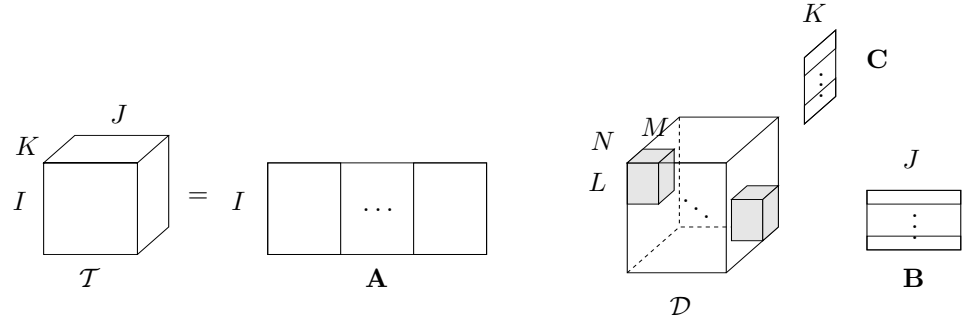


FIG. 2.3. *Interpretation of decomposition* (2.11) *in terms of the multiplication of a block-diagonal core tensor $\mathcal{D}$ by transformation matrices* **A**, **B**, *and* **C**.

2.3. Type-2 decomposition in rank-$(L, M, \cdot)$ terms.

DEFINITION 2.4. *A* type-2 decomposition of a tensor $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ in a sum of rank-$(L, M, \cdot)$ terms *is a decomposition of $\mathcal{T}$ of the form*

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{C}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r, \tag{2.16}$$

in which $\mathcal{C}_r \in \mathbb{K}^{L\times M\times K}$ *(with mode-*1 *rank equal to* L *and mode-*2 *rank equal to* M*) and in which* $\mathbf{A}_r \in \mathbb{K}^{I\times L}$ *(with* $I \geqslant L$*) and* $\mathbf{B}_r \in \mathbb{K}^{J\times M}$ *(with* $J \geqslant M$*) are full column rank,* $1 \leqslant r \leqslant R$*.*

Remark 2. The label "type 2" is reminiscent of the term "Tucker-2 decomposition."

Remark 3. One could also consider a type-2 decomposition in rank-$(L_r, M_r, \cdot)$ terms, where the different terms possibly have different mode-1 and/or mode-2 rank. In this paper we focus on the decomposition in rank-$(L, M, \cdot)$ terms.

Define partitioned matrices $\mathbf{A} = [\mathbf{A}_1 \dots \mathbf{A}_R]$ and $\mathbf{B} = [\mathbf{B}_1 \dots \mathbf{B}_R]$. In terms of the standard matrix representations of $\mathcal{T}$, (2.16) can be written as

$$\mathbf{T}_{IJ\times K} = (\mathbf{A} \odot \mathbf{B}) \cdot \begin{pmatrix} (\mathcal{C}_1)_{(LM\times K)} \\ \vdots \\ (\mathcal{C}_R)_{(LM\times K)} \end{pmatrix}, \tag{2.17}$$

$$\mathbf{T}_{JK\times I} = [(\mathcal{C}_1 \bullet_2 \mathbf{B}_1)_{JK\times L} \ \dots \ (\mathcal{C}_R \bullet_2 \mathbf{B}_R)_{JK\times L}] \cdot \mathbf{A}^T, \tag{2.18}$$

$$\mathbf{T}_{KI\times J} = [(\mathcal{C}_1 \bullet_1 \mathbf{A}_1)_{KI\times M} \ \dots \ (\mathcal{C}_R \bullet_1 \mathbf{A}_R)_{KI\times M}] \cdot \mathbf{B}^T. \tag{2.19}$$

Define $\mathcal{C} \in \mathbb{K}^{LR\times MR\times K}$ as an all-zero tensor, except for the entries given by

$$(\mathcal{C})_{(r-1)L+l,(r-1)M+m,k} = (\mathcal{C}_r)_{lmk} \quad \forall l, m, k, r.$$

Then (2.16) can also be written as

$$\mathcal{T} = \mathcal{C} \bullet_1 \mathbf{A} \bullet_2 \mathbf{B}.$$

It is clear that in (2.16) one can arbitrarily permute the different terms. Also, one can postmultiply $\mathbf{A}_r$ by a nonsingular matrix $\mathbf{F}_r \in \mathbb{K}^{L\times L}$ and postmultiply $\mathbf{B}_r$ by a nonsingular matrix $\mathbf{G}_r \in \mathbb{K}^{M\times M}$, provided $\mathcal{C}_r$ is replaced by $\mathcal{C}_r \bullet_1 (\mathbf{F}_r)^{-1} \bullet_2 (\mathbf{G}_r)^{-1}$. We call the decomposition essentially unique when it is subject only to these trivial indeterminacies. Two representations $(\mathbf{A}, \mathbf{B}, \mathcal{C})$ and $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathcal{C}})$ that are the same up to trivial indeterminacies are called essentially equal. We can (partially) normalize (2.16) by representing each term by its normalized Tucker-2 decomposition. The decomposition is visualized in Figure 2.4.

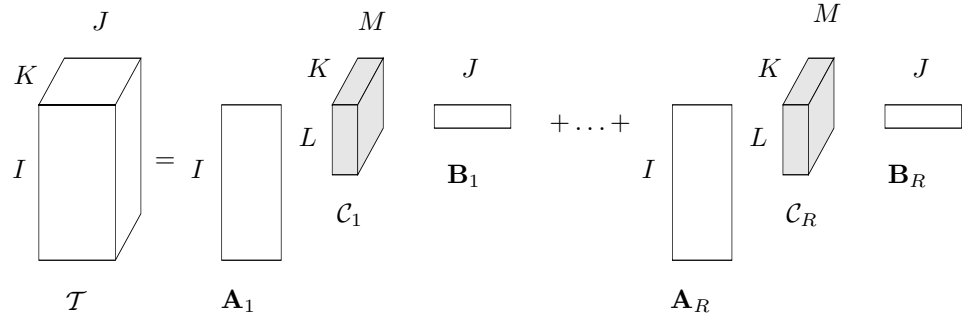


FIG. 2.4. *Visualization of the type-*2 *decomposition of a tensor in a sum of rank-*$(L, M, \cdot)$ *terms.*

3. Basic lemmas. In this section we list a number of lemmas that we will use in the analysis of the uniqueness of the block term decompositions.

Let $\omega(\mathbf{x})$ denote the number of nonzero entries of a vector $\mathbf{x}$. The following lemma was originally proposed by Kruskal in [38]. It is known as the *permutation lemma.*

It plays a crucial role in the proof of (1.15). The proof was reformulated in terms of accessible basic linear algebra in [54]. An alternative proof was given in [31]. The link between the two proofs is also discussed in [54].

LEMMA 3.1 (permutation lemma). *Consider two matrices* $\bar{\mathbf{A}}$, $\mathbf{A} \in \mathbb{K}^{I \times R}$, *that have no zero columns. If for every vector* $\mathbf{x}$ *such that* $\omega(\mathbf{x}^T\bar{\mathbf{A}}) \leqslant R - r_{\bar{\mathbf{A}}} + 1$, *we have* $\omega(\mathbf{x}^T\mathbf{A}) \leqslant \omega(\mathbf{x}^T\bar{\mathbf{A}})$, *then there exists a unique permutation matrix* $\mathbf{\Pi}$ *and a unique nonsingular diagonal matrix* $\mathbf{\Lambda}$ *such that* $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}$.

In [19] we have introduced a generalization of the permutation lemma to partitioned matrices. Let us first introduce some additional prerequisites. Let $\omega'(\mathbf{x})$ denote the number of parts of a partitioned vector $\mathbf{x}$ that are not all-zero. We call the partitioning of a partitioned matrix $\mathbf{A}$ uniform when all submatrices are of the same size. Finally, we generalize the k-rank concept to partitioned matrices [19].

DEFINITION 3.2. *The* k'-rank *of a (not necessarily uniformly) partitioned matrix* $\mathbf{A}$, *denoted by* $\text{rank}_{k'}(\mathbf{A})$ *or* $k'_{\mathbf{A}}$, *is the maximal number* r *such that any set of* r *submatrices of* $\mathbf{A}$ *yields a set of linearly independent columns.*

Let $\mathbf{A} \in \mathbb{K}^{I \times LR}$ be uniformly partitioned in R matrices $\mathbf{A}_r \in \mathbb{K}^{I \times L}$. Generically, we have $k'_{\mathbf{A}} = \min(\lfloor \frac{I}{L} \rfloor, R)$.

We are now in a position to formulate the lemma that generalizes the permutation lemma.

LEMMA 3.3 (equivalence lemma for partitioned matrices). *Consider* $\bar{\mathbf{A}}$, $\mathbf{A} \in \mathbb{K}^{I \times \sum_{r=1}^{R} L_r}$, *partitioned in the same but not necessarily uniform way into* R *submatrices that are full column rank. Suppose that for every* $\mu \leqslant R - k'_{\bar{\mathbf{A}}} + 1$ *there holds that for a generic*[1] *vector* $\mathbf{x}$ *such that* $\omega'(\mathbf{x}^T\bar{\mathbf{A}}) \leqslant \mu$, *we have* $\omega'(\mathbf{x}^T\mathbf{A}) \leqslant \omega'(\mathbf{x}^T\bar{\mathbf{A}})$. *Then there exists a unique block-permutation matrix* $\mathbf{\Pi}$ *and a unique nonsingular block-diagonal matrix* $\mathbf{\Lambda}$, *such that* $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}$, *where the block-transformation is compatible with the block-structure of* $\mathbf{A}$ *and* $\bar{\mathbf{A}}$.

(Compared to the presentation in [19] we have dropped the irrelevant complex conjugation of $\mathbf{x}$.)

We note that the rank $r_{\bar{\mathbf{A}}}$ in the permutation lemma has been replaced by the k'-rank $k'_{\bar{\mathbf{A}}}$ in Lemma 3.3. The reason is that the permutation lemma admits a simpler proof when we can assume that $r_{\bar{\mathbf{A}}} = k_{\bar{\mathbf{A}}}$. It is this simpler proof, given in [31], that is generalized in [19].

The following lemma gives a lower-bound on the k'-rank of a Khatri–Rao product of partitioned matrices [19].

LEMMA 3.4. *Consider partitioned matrices* $\mathbf{A} = [\mathbf{A}_1 \ \ldots \ \mathbf{A}_R]$ *with* $\mathbf{A}_r \in \mathbb{K}^{I \times L_r}$, $1 \leqslant r \leqslant R$, *and* $\mathbf{B} = [\mathbf{B}_1 \ \ldots \ \mathbf{B}_R]$ *with* $\mathbf{B}_r \in \mathbb{K}^{J \times M_r}$, $1 \leqslant r \leqslant R$.

(i) *If* $k'_{\mathbf{A}} = 0$ *or* $k'_{\mathbf{B}} = 0$, *then* $k'_{\mathbf{A} \odot \mathbf{B}} = 0$.

(ii) *If* $k'_{\mathbf{A}} \geqslant 1$ *and* $k'_{\mathbf{B}} \geqslant 1$, *then* $k'_{\mathbf{A} \odot \mathbf{B}} \geqslant \min(k'_{\mathbf{A}} + k'_{\mathbf{B}} - 1, R)$.

Finally, we have a lemma that says that a Khatri–Rao product of partitioned matrices is generically full column rank [19].

[1] We mean the following. Consider, for instance, a partitioned matrix $\bar{\mathbf{A}} = [\mathbf{a}_1 \ \mathbf{a}_2 | \mathbf{a}_3 \ \mathbf{a}_4] \in \mathbb{K}^{4 \times 4}$ that is full column rank. The set $\mathsf{S} = \{\mathbf{x} | \omega'(\mathbf{x}^T\bar{\mathbf{A}}) \leqslant 1\}$ is the union of two subspaces, S_1 and S_2, consisting of the set of vectors orthogonal to $\{\mathbf{a}_1, \mathbf{a}_2\}$ and $\{\mathbf{a}_3, \mathbf{a}_4\}$, respectively. When we say that for a generic vector $\mathbf{x}$ such that $\omega'(\mathbf{x}^T\bar{\mathbf{A}}) \leqslant 1$, we have $\omega'(\mathbf{x}^T\mathbf{A}) \leqslant \omega'(\mathbf{x}^T\bar{\mathbf{A}})$, we mean that $\omega'(\mathbf{x}^T\mathbf{A}) \leqslant \omega'(\mathbf{x}^T\bar{\mathbf{A}})$ holds with probability one for a vector $\mathbf{x}$ drawn from a continuous probability density function over S_1 and that $\omega'(\mathbf{x}^T\mathbf{A}) \leqslant \omega'(\mathbf{x}^T\bar{\mathbf{A}})$ also holds with probability one for a vector $\mathbf{x}$ drawn from a continuous probability density function over S_2. In general, the set $\mathsf{S} = \{\mathbf{x} | \omega'(\mathbf{x}^T\bar{\mathbf{A}}) \leqslant \mu\}$ consists of a finite union of subspaces, where we count only the subspaces that are not contained in an other subspace. For each of these subspaces, the property should hold with probability one for a vector $\mathbf{x}$ drawn from a continuous probability density function over that subspace.

LEMMA 3.5. *Consider partitioned matrices* $\mathbf{A} = [\mathbf{A}_1 \ \ldots \ \mathbf{A}_R]$ *with* $\mathbf{A}_r \in \mathbb{K}^{I \times L_r}$, $1 \leqslant r \leqslant R$, *and* $\mathbf{B} = [\mathbf{B}_1 \ \ldots \ \mathbf{B}_R]$ *with* $\mathbf{B}_r \in \mathbb{K}^{J \times M_r}$, $1 \leqslant r \leqslant R$. *Generically we have that* $\operatorname{rank}(\mathbf{A} \odot \mathbf{B}) = \min(IJ, \sum_{r=1}^{R} L_r M_r)$.

4. The decomposition in rank-$(L_r, L_r, 1)$ terms. In this section we derive several conditions under which essential uniqueness of the decomposition in rank-$(L, L, 1)$ or rank-$(L_r, L_r, 1)$ terms is guaranteed. We use the notation introduced in section 2.1.

For decompositions in generic rank-$(L, L, 1)$ terms, the results of this section can be summarized as follows. We have essential uniqueness if

(i) Theorem 4.1:

$$\min(I, J) \geqslant LR \quad \text{and} \quad \mathbf{C} \text{ does not have proportional columns;} \tag{4.1}$$

(ii) Theorem 4.4:

$$K \geqslant R \quad \text{and} \quad \min\left(\left\lfloor \frac{I}{L} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{J}{L} \right\rfloor, R\right) \geqslant R + 2; \tag{4.2}$$

(iii) Theorem 4.5:

$$I \geqslant LR \quad \text{and} \quad \min\left(\left\lfloor \frac{J}{L} \right\rfloor, R\right) + \min(K, R) \geqslant R + 2 \tag{4.3}$$

or

$$J \geqslant LR \quad \text{and} \quad \min\left(\left\lfloor \frac{I}{L} \right\rfloor, R\right) + \min(K, R) \geqslant R + 2; \tag{4.4}$$

(iv) Theorem 4.7:

$$\left\lfloor \frac{IJ}{L^2} \right\rfloor \geqslant R \quad \text{and} \quad \min\left(\left\lfloor \frac{I}{L} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{J}{L} \right\rfloor, R\right) + \min(K, R) \geqslant 2R + 2. \tag{4.5}$$

First we mention a result of which the first version appeared, in a slightly different form, in [52]. The proof describes a procedure by which, under the given conditions, the components of the decomposition may be computed. This procedure is a generalization of the computation of PARAFAC from the generalized eigenvectors of the pencil $(\mathbf{T}_{I \times J,1}^T, \mathbf{T}_{I \times J,2}^T)$, as explained in [20, section 1.4].

THEOREM 4.1. *Let* $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ *represent a decomposition of* $\mathcal{T}$ *in rank-*$(L_r, L_r, 1)$ *terms,* $1 \leqslant r \leqslant R$. *Suppose that* $\mathbf{A}$ *and* $\mathbf{B}$ *are full column rank and that* $\mathbf{C}$ *does not have proportional columns. Then* $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ *is essentially unique.*

Proof. Assume that $c_{21}, \ldots, c_{2R}$ are different from zero and that c_{11}/c_{21}, $\ldots$, c_{1R}/c_{2R} are mutually different. (If this is not the case, consider linear combinations of matrix slices in the reasoning below.) From (2.9) we have

$$\mathbf{T}_{I \times J,1} = \mathbf{A} \cdot \operatorname{blockdiag}(c_{11}\mathbf{I}_{L_1 \times L_1}, \ldots, c_{1R}\mathbf{I}_{L_R \times L_R}) \cdot \mathbf{B}^T, \tag{4.6}$$

$$\mathbf{T}_{I \times J,2} = \mathbf{A} \cdot \operatorname{blockdiag}(c_{21}\mathbf{I}_{L_1 \times L_1}, \ldots, c_{2R}\mathbf{I}_{L_R \times L_R}) \cdot \mathbf{B}^T. \tag{4.7}$$

This means that the columns of $(\mathbf{A}^T)^\dagger$ are generalized eigenvectors of the pencil $(\mathbf{T}_{I \times J,1}^T, \mathbf{T}_{I \times J,2}^T)$ [4, 22]. The columns of the rth submatrix of $\mathbf{A}$ are associated with the same generalized eigenvalue c_{1r}/c_{2r} and can therefore not be separated, $1 \leqslant r \leqslant R$. This is consistent with the indeterminacies of the decomposition. On the other

hand, the different submatrices of $\mathbf{A}$ can be separated, as they correspond to different generalized eigenvalues. After computation of a possible matrix $\mathbf{A}$, the corresponding matrix $\mathbf{B}$ can be computed, up to scaling of its submatrices, from (4.7):

$$(\mathbf{A}^{\dagger} \cdot \mathbf{T}_{I\times J,2})^T = \mathbf{B} \cdot \text{blockdiag}(c_{21}\mathbf{I}_{L_1\times L_1}, \ldots, c_{2R}\mathbf{I}_{L_R\times L_R}).$$

Matrix $\mathbf{C}$ finally follows from (2.4):

$$\mathbf{C} = \left\{[(\mathbf{A}_1 \odot_c \mathbf{B}_1)\mathbf{1}_{L_1} \ \ldots \ (\mathbf{A}_R \odot_c \mathbf{B}_R)\mathbf{1}_{L_R}]^{\dagger} \cdot \mathbf{T}_{IJ\times K}\right\}^T. \qquad \square$$

Next, we derive generalizations of Kruskal's condition (1.15) under which essential uniqueness of $\mathbf{A}$, or $\mathbf{B}$, or $\mathbf{C}$ is guaranteed. Lemma 4.2 concerns essential uniqueness of $\mathbf{C}$. In its proof, we assume that the partitioning of $\mathbf{A}$ and $\mathbf{B}$ is uniform. Hence, the lemma applies only to the decomposition in rank-$(L, L, 1)$ terms. Lemma 4.3 concerns essential uniqueness of $\mathbf{A}$ and/or $\mathbf{B}$. This lemma applies more generally to the decomposition in rank-$(L_r, L_r, 1)$ terms. Later in this section, essential uniqueness of the decomposition of $\mathcal{T}$ will be inferred from essential uniqueness of one or more of the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$.

LEMMA 4.2. *Let* $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ *represent a decomposition of* $\mathcal{T}$ *in* R *rank-*$(L, L, 1)$ *terms. Suppose the condition*

$$k'_{\mathbf{A}} + k'_{\mathbf{B}} + k_{\mathbf{C}} \geqslant 2R + 2 \tag{4.8}$$

holds and that we have an alternative decomposition of $\mathcal{T}$*, represented by* $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$*. Then there holds* $\bar{\mathbf{C}} = \mathbf{C} \cdot \mathbf{\Pi}_c \cdot \mathbf{\Lambda}_c$*, in which* $\mathbf{\Pi}_c$ *is a permutation matrix and* $\mathbf{\Lambda}_c$ *a nonsingular diagonal matrix.*

Proof. We work in analogy with [54]. Equality of $\mathbf{C}$ and $\bar{\mathbf{C}}$, up to column permutation and scaling, follows from the permutation lemma if we can prove that for any $\mathbf{x}$ such that $\omega(\mathbf{x}^T\bar{\mathbf{C}}) \leqslant R - r_{\bar{\mathbf{C}}} + 1$, there holds $\omega(\mathbf{x}^T\mathbf{C}) \leqslant \omega(\mathbf{x}^T\bar{\mathbf{C}})$. This proof is structured as follows. First, we derive an upper-bound on $\omega(\mathbf{x}^T\bar{\mathbf{C}})$. Then we derive a lower-bound on $\omega(\mathbf{x}^T\bar{\mathbf{C}})$. Combination of the two bounds yields the desired result.

(i) *Derivation of an upper-bound on* $\omega(\mathbf{x}^T\bar{\mathbf{C}})$. From (2.9) we have that $\text{vec}(\mathbf{T}^T_{I\times J,k}) = [(\mathbf{A}_1 \odot_c \mathbf{B}_1)\mathbf{1}_L \ \ldots \ (\mathbf{A}_R \odot_c \mathbf{B}_R)\mathbf{1}_L] \cdot [c_{k1} \ldots c_{kR}]^T$. Consider the linear combination of $(I \times J)$ slices $\sum_{k=1}^K x_k \mathbf{T}_{I\times J,k}$. Since $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$ both represent a decomposition of $\mathcal{T}$, we have

$$\begin{aligned}&[(\mathbf{A}_1 \odot_c \mathbf{B}_1)\mathbf{1}_L \ \ldots \ (\mathbf{A}_R \odot_c \mathbf{B}_R)\mathbf{1}_L] \cdot \mathbf{C}^T\mathbf{x} \\ &\quad = [(\bar{\mathbf{A}}_1 \odot_c \bar{\mathbf{B}}_1)\mathbf{1}_L \ \ldots \ (\bar{\mathbf{A}}_R \odot_c \bar{\mathbf{B}}_R)\mathbf{1}_L] \cdot \bar{\mathbf{C}}^T\mathbf{x}.\end{aligned}$$

By Lemma 3.4, the matrix $\mathbf{A} \odot \mathbf{B}$ has full column rank. The matrix $[(\mathbf{A}_1 \odot_c \mathbf{B}_1)\mathbf{1}_L \ \ldots \ (\mathbf{A}_R \odot_c \mathbf{B}_R)\mathbf{1}_L]$ is equal to $(\mathbf{A} \odot \mathbf{B}) \cdot [\text{vec}(\mathbf{I}_{L\times L})^T \ \ldots \ \text{vec}(\mathbf{I}_{L\times L})^T]^T$ and thus also has full column rank. This implies that if $\omega(\mathbf{x}^T\bar{\mathbf{C}}) = 0$, then also $\omega(\mathbf{x}^T\mathbf{C}) = 0$. Hence, $\text{null}(\bar{\mathbf{C}}) \subseteq \text{null}(\mathbf{C})$. Basic matrix algebra yields $\text{span}(\mathbf{C}) \subseteq \text{span}(\bar{\mathbf{C}})$ and $r_{\mathbf{C}} \leqslant r_{\bar{\mathbf{C}}}$. This implies that if $\omega(\mathbf{x}^T\bar{\mathbf{C}}) \leqslant R - r_{\bar{\mathbf{C}}} + 1$, then

$$\omega(\mathbf{x}^T\bar{\mathbf{C}}) \leqslant R - r_{\bar{\mathbf{C}}} + 1 \leqslant R - r_{\mathbf{C}} + 1 \leqslant R - k_{\mathbf{C}} + 1 \leqslant k'_{\mathbf{A}} + k'_{\mathbf{B}} - (R + 1), \tag{4.9}$$

where the last inequality corresponds to condition (4.8).

(ii) *Derivation of a lower-bound on* $\omega(\mathbf{x}^T\bar{\mathbf{C}})$. By (2.9), the linear combination of $(I \times J)$ slices $\sum_{k=1}^K x_k \mathbf{T}_{I\times J,k}$ is given by

$$\begin{aligned}&\mathbf{A} \cdot \text{blockdiag}(\mathbf{x}^T\mathbf{c}_1\mathbf{I}_{L\times L}, \ldots, \mathbf{x}^T\mathbf{c}_R\mathbf{I}_{L\times L}) \cdot \mathbf{B}^T \\ &\quad = \bar{\mathbf{A}} \cdot \text{blockdiag}(\mathbf{x}^T\bar{\mathbf{c}}_1\mathbf{I}_{L\times L}, \ldots, \mathbf{x}^T\bar{\mathbf{c}}_R\mathbf{I}_{L\times L}) \cdot \bar{\mathbf{B}}^T.\end{aligned}$$

We have

$$\begin{aligned} L\omega(\mathbf{x}^T\bar{\mathbf{C}}) &= r_{\text{blockdiag}(\mathbf{x}^T\bar{\mathbf{c}}_1\mathbf{I}_{L\times L},\ldots,\mathbf{x}^T\bar{\mathbf{c}}_R\mathbf{I}_{L\times L})} \\ &\geqslant r_{\bar{\mathbf{A}}\cdot\text{blockdiag}(\mathbf{x}^T\bar{\mathbf{c}}_1\mathbf{I}_{L\times L},\ldots,\mathbf{x}^T\bar{\mathbf{c}}_R\mathbf{I}_{L\times L})\cdot\bar{\mathbf{B}}^T} \\ &= r_{\mathbf{A}\cdot\text{blockdiag}(\mathbf{x}^T\mathbf{c}_1\mathbf{I}_{L\times L},\ldots,\mathbf{x}^T\mathbf{c}_R\mathbf{I}_{L\times L})\cdot\mathbf{B}^T}. \end{aligned} \tag{4.10}$$

Let $\gamma = \omega(\mathbf{x}^T\mathbf{C})$ and let $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ consist of the submatrices of $\mathbf{A}$ and $\mathbf{B}$, respectively, corresponding to the nonzero elements of $\mathbf{x}^T\mathbf{C}$. Then $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ both have γL columns. Let $\mathbf{u}$ be the $(\gamma \times 1)$ vector containing the nonzero elements of $\mathbf{x}^T\mathbf{C}$ such that

$$\mathbf{A}\cdot\text{blockdiag}(\mathbf{x}^T\mathbf{c}_1\mathbf{I}_{L\times L},\ldots,\mathbf{x}^T\mathbf{c}_R\mathbf{I}_{L\times L})\cdot\mathbf{B}^T = \tilde{\mathbf{A}}\cdot\text{blockdiag}(u_1\mathbf{I}_{L\times L},\ldots,u_\gamma\mathbf{I}_{L\times L})\cdot\tilde{\mathbf{B}}^T.$$

Sylvester's inequality now yields

$$\begin{aligned} r_{\mathbf{A}\cdot\text{blockdiag}(\mathbf{x}^T\mathbf{c}_1\mathbf{I}_{L\times L},\ldots,\mathbf{x}^T\mathbf{c}_R\mathbf{I}_{L\times L})\cdot\mathbf{B}^T} &= r_{\tilde{\mathbf{A}}\cdot\text{blockdiag}(u_1\mathbf{I}_{L\times L},\ldots,u_\gamma\mathbf{I}_{L\times L})\cdot\tilde{\mathbf{B}}^T} \\ &\geqslant r_{\tilde{\mathbf{A}}} + r_{\text{blockdiag}(u_1\mathbf{I}_{L\times L},\ldots,u_\gamma\mathbf{I}_{L\times L})\cdot\tilde{\mathbf{B}}^T} - \gamma L \\ &= r_{\tilde{\mathbf{A}}} + r_{\tilde{\mathbf{B}}} - \gamma L, \end{aligned} \tag{4.11}$$

where the last equality is due to the fact that $\mathbf{u}$ has no zero elements. From the definition of k′-rank, we have

$$r_{\tilde{\mathbf{A}}} \geqslant L\min(\gamma, k'_{\mathbf{A}}), \quad r_{\tilde{\mathbf{B}}} \geqslant L\min(\gamma, k'_{\mathbf{B}}). \tag{4.12}$$

Combination of (4.10)–(4.12) yields the following lower-bound on $\omega(\mathbf{x}^T\bar{\mathbf{C}})$:

$$\omega(\mathbf{x}^T\bar{\mathbf{C}}) \geqslant \min(\gamma, k'_{\mathbf{A}}) + \min(\gamma, k'_{\mathbf{B}}) - \gamma. \tag{4.13}$$

(iii) *Combination of the two bounds.* Combination of (4.9) and (4.13) yields

$$\min(\gamma, k'_{\mathbf{A}}) + \min(\gamma, k'_{\mathbf{B}}) - \gamma \leqslant \omega(\mathbf{x}^T\bar{\mathbf{C}}) \leqslant k'_{\mathbf{A}} + k'_{\mathbf{B}} - (R+1). \tag{4.14}$$

To be able to apply the permutation lemma, we need to show that $\gamma = \omega(\mathbf{x}^T\mathbf{C}) \leqslant \omega(\mathbf{x}^T\bar{\mathbf{C}})$. By (4.14), it suffices to show that $\gamma < \min(k'_{\mathbf{A}}, k'_{\mathbf{B}})$. We prove this by contradiction. Suppose $\gamma > \max(k'_{\mathbf{A}}, k'_{\mathbf{B}})$. Then (4.14) yields $\gamma \geqslant R+1$, which is impossible. Suppose next that $k'_{\mathbf{A}} \leqslant \gamma \leqslant k'_{\mathbf{B}}$. Then (4.14) yields $k'_{\mathbf{B}} \geqslant R+1$, which is also impossible. Since $\mathbf{A}$ and $\mathbf{B}$ can be exchanged in the latter case, we have that $\gamma < \min(k'_{\mathbf{A}}, k'_{\mathbf{B}})$. Equation (4.14) now implies that $\omega(\mathbf{x}^T\mathbf{C}) \leqslant \omega(\mathbf{x}^T\bar{\mathbf{C}})$. By the permutation lemma, there exist a unique permutation matrix $\mathbf{\Pi}_c$ and a nonsingular diagonal matrix $\mathbf{\Lambda}_c$ such that $\bar{\mathbf{C}} = \mathbf{C}\cdot\mathbf{\Pi}_c\cdot\mathbf{\Lambda}_c$. □

In the following lemma, we prove essential uniqueness of $\mathbf{A}$ and $\mathbf{B}$ when we restrict our attention to alternative $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ that are, in some sense, "nonsingular." What we mean is that there are no linear dependencies between columns that are not imposed by the dimensionality constraints.

LEMMA 4.3. *Let* $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ *represent a decomposition of* $\mathcal{T}$ *in rank-*$(L_r, L_r, 1)$ *terms,* $1 \leqslant r \leqslant R$. *Suppose the condition*

$$k'_{\mathbf{A}} + k'_{\mathbf{B}} + k_{\mathbf{C}} \geqslant 2R + 2 \tag{4.15}$$

holds and that we have an alternative decomposition of $\mathcal{T}$, *represented by* $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$, *with* $k'_{\bar{\mathbf{A}}}$ *and* $k'_{\bar{\mathbf{B}}}$ *maximal under the given dimensionality constraints. Then there holds* $\bar{\mathbf{A}} = \mathbf{A}\cdot\mathbf{\Pi}_a\cdot\mathbf{\Lambda}_a$, *in which* $\mathbf{\Pi}_a$ *is a block permutation matrix and* $\mathbf{\Lambda}_a$ *a square*

nonsingular block-diagonal matrix, compatible with the block structure of $\mathbf{A}$*. There also holds* $\bar{\mathbf{B}} = \mathbf{B} \cdot \mathbf{\Pi}_b \cdot \mathbf{\Lambda}_b$*, in which* $\mathbf{\Pi}_b$ *is a block permutation matrix and* $\mathbf{\Lambda}_b$ *a square nonsingular block-diagonal matrix, compatible with the block structure of* $\mathbf{B}$*.*

Proof. It suffices to prove the lemma for $\mathbf{A}$. The result for $\mathbf{B}$ can be obtained by switching modes. We work in analogy with the proof of Lemma 4.2. Essential uniqueness of $\mathbf{A}$ now follows from the equivalence lemma for partitioned matrices.

(i) *Derivation of an upper-bound on* $\omega'(\mathbf{x}^T\bar{\mathbf{A}})$*.* The constraint on $k'_{\bar{\mathbf{A}}}$ implies that $k'_{\bar{\mathbf{A}}} \geqslant k'_{\mathbf{A}}$. Hence, if $\omega'(\mathbf{x}^T\bar{\mathbf{A}}) \leqslant R - k'_{\bar{\mathbf{A}}} + 1$, then

$$\omega'(\mathbf{x}^T\bar{\mathbf{A}}) \leqslant R - k'_{\bar{\mathbf{A}}} + 1 \leqslant R - k'_{\mathbf{A}} + 1 \leqslant k'_{\mathbf{B}} + k_{\mathbf{C}} - (R+1), \tag{4.16}$$

where the last inequality corresponds to condition (4.15).

(ii) *Derivation of a lower-bound on* $\omega'(\mathbf{x}^T\bar{\mathbf{A}})$*.* By (2.7), the linear combination of $(J \times K)$ slices $\sum_{i=1}^{I} x_i \mathbf{T}_{J\times K,i}$ is given by

$$\mathbf{B} \cdot \text{blockdiag}(\mathbf{A}_1^T\mathbf{x}, \ldots, \mathbf{A}_R^T\mathbf{x}) \cdot \mathbf{C}^T = \bar{\mathbf{B}} \cdot \text{blockdiag}(\bar{\mathbf{A}}_1^T\mathbf{x}, \ldots, \bar{\mathbf{A}}_R^T\mathbf{x}) \cdot \bar{\mathbf{C}}^T.$$

We have

$$\begin{aligned} \omega'(\mathbf{x}^T\bar{\mathbf{A}}) &= r_{\text{blockdiag}(\bar{\mathbf{A}}_1^T\mathbf{x},\ldots,\bar{\mathbf{A}}_R^T\mathbf{x})} \\ &\geqslant r_{\bar{\mathbf{B}}\cdot\text{blockdiag}(\bar{\mathbf{A}}_1^T\mathbf{x},\ldots,\bar{\mathbf{A}}_R^T\mathbf{x})\cdot\bar{\mathbf{C}}^T} \\ &= r_{\mathbf{B}\cdot\text{blockdiag}(\mathbf{A}_1^T\mathbf{x},\ldots,\mathbf{A}_R^T\mathbf{x})\cdot\mathbf{C}^T}. \end{aligned} \tag{4.17}$$

Let $\gamma = \omega'(\mathbf{x}^T\mathbf{A})$ and let $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{C}}$ consist of the submatrices of $\mathbf{B}\cdot\text{blockdiag}(\mathbf{A}_1^T\mathbf{x}, \ldots, \mathbf{A}_R^T\mathbf{x})$ and $\mathbf{C}$, respectively, corresponding to the parts of $\mathbf{x}^T\mathbf{A}$ that are not all-zero. Then $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{C}}$ both have γ columns. Sylvester's inequality now yields

$$r_{\mathbf{B}\cdot\text{blockdiag}(\mathbf{A}_1^T\mathbf{x},\ldots,\mathbf{A}_R^T\mathbf{x})\cdot\mathbf{C}^T} \geqslant r_{\tilde{\mathbf{B}}} + r_{\tilde{\mathbf{C}}} - \gamma. \tag{4.18}$$

The matrix $\tilde{\mathbf{B}}$ consists of γ nonzero vectors, sampled in the column spaces of the submatrices of $\mathbf{B}$ that correspond to the parts of $\mathbf{x}^T\mathbf{A}$ that are not all-zero. From the definition of k′-rank, we have

$$r_{\tilde{\mathbf{B}}} \geqslant \min(\gamma, k'_{\mathbf{B}}). \tag{4.19}$$

On the other hand, from the definition of k-rank, we have

$$r_{\tilde{\mathbf{C}}} \geqslant \min(\gamma, k_{\mathbf{C}}). \tag{4.20}$$

Combination of (4.17)–(4.20) yields the following lower-bound on $\omega'(\mathbf{x}^T\bar{\mathbf{A}})$:

$$\omega'(\mathbf{x}^T\bar{\mathbf{A}}) \geqslant \min(\gamma, k'_{\mathbf{B}}) + \min(\gamma, k_{\mathbf{C}}) - \gamma. \tag{4.21}$$

(iii) *Combination of the two bounds.* This is analogous to Lemma 4.2. □

We now use Lemmas 4.2 and 4.3, which concern the essential uniqueness of the individual matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, to establish essential uniqueness of the overall decomposition of $\mathcal{T}$. Theorem 4.4 states that if $\mathbf{C}$ is full column rank and tall (meaning that $R \leqslant K$), then its essential uniqueness implies essential uniqueness of the overall tensor decomposition. Theorem 4.5 is the equivalent for $\mathbf{A}$ (or $\mathbf{B}$). However, none of

the factor matrices needs to be tall for the decomposition to be unique. A more general case is dealt with in Theorem 4.7. Its proof makes use of Lemma 4.6, guaranteeing that under a generalized Kruskal condition, $\mathbf{A}$ and $\mathbf{B}$ not only are individually essentially unique but, moreover, are subject to the same permutation of their submatrices.

We first consider essential uniqueness of a tall full column rank matrix $\mathbf{C}$.

THEOREM 4.4. *Let* $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ *represent a decomposition of* $\mathcal{T}$ *in* R *rank-*$(L, L, 1)$ *terms. Suppose that we have an alternative decomposition of* $\mathcal{T}$*, represented by* $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$*. If*

$$k_{\mathbf{C}} = R \quad \textit{and} \quad k'_{\mathbf{A}} + k'_{\mathbf{B}} \geqslant R + 2, \tag{4.22}$$

then $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ *and* $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$ *are essentially equal.*

Proof. From (2.4) we have

$$\begin{aligned} \mathbf{T}_{IJ\times K} &= [(\mathbf{A}_1 \odot_c \mathbf{B}_1)\mathbf{1}_L \ \ldots \ (\mathbf{A}_R \odot_c \mathbf{B}_R)\mathbf{1}_L] \cdot \mathbf{C}^T \\ &= [(\bar{\mathbf{A}}_1 \odot_c \bar{\mathbf{B}}_1)\mathbf{1}_L \ \ldots \ (\bar{\mathbf{A}}_R \odot_c \bar{\mathbf{B}}_R)\mathbf{1}_L] \cdot \bar{\mathbf{C}}^T. \end{aligned} \tag{4.23}$$

From Lemma 4.2 we have

$$\bar{\mathbf{C}} = \mathbf{C} \cdot \mathbf{\Pi}_c \cdot \mathbf{\Lambda}_c. \tag{4.24}$$

Since $k_{\mathbf{C}} = R$, $\mathbf{C}$ is full column rank. Substitution of (4.24) in (4.23) now yields

$$\begin{aligned} &[(\mathbf{A}_1 \odot_c \mathbf{B}_1)\mathbf{1}_L \ \ldots \ (\mathbf{A}_R \odot_c \mathbf{B}_R)\mathbf{1}_L] \\ &\quad = [(\bar{\mathbf{A}}_1 \odot_c \bar{\mathbf{B}}_1)\mathbf{1}_L \ \ldots \ (\bar{\mathbf{A}}_R \odot_c \bar{\mathbf{B}}_R)\mathbf{1}_L] \cdot \mathbf{\Lambda}_c^T \cdot \mathbf{\Pi}_c^T. \end{aligned} \tag{4.25}$$

Taking into account that $(\bar{\mathbf{A}}_r \odot_c \bar{\mathbf{B}}_r)\mathbf{1}_L$ is a vector representation of the matrix $\bar{\mathbf{A}}_r \cdot \bar{\mathbf{B}}_r^T$, $1 \leqslant r \leqslant R$, this implies that the matrices $\bar{\mathbf{A}}_r \cdot \bar{\mathbf{B}}_r^T$ are ordered in the same way as the vectors $\bar{\mathbf{c}}_r$. Furthermore, if $\bar{\mathbf{c}}_i = \lambda \mathbf{c}_j$, then $(\bar{\mathbf{A}}_i \odot_c \bar{\mathbf{B}}_i)\mathbf{1}_L = \lambda^{-1}(\mathbf{A}_j \odot_c \mathbf{B}_j)\mathbf{1}_L$, or, equivalently, $\bar{\mathbf{A}}_i \cdot \bar{\mathbf{B}}_i^T = \lambda^{-1}\mathbf{A}_j \cdot \mathbf{B}_j^T$. ∎

We now consider essential uniqueness of a tall full column rank matrix $\mathbf{A}$ or $\mathbf{B}$.

THEOREM 4.5. *Let* $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ *represent a decomposition of* $\mathcal{T}$ *in rank-*$(L_r, L_r, 1)$ *terms,* $1 \leqslant r \leqslant R$*. Suppose that we have an alternative decomposition of* $\mathcal{T}$*, represented by* $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$*, with* $k'_{\bar{\mathbf{A}}}$ *and* $k'_{\bar{\mathbf{B}}}$ *maximal under the given dimensionality constraints. If*

$$k'_{\mathbf{A}} = R \quad \textit{and} \quad k'_{\mathbf{B}} + k_{\mathbf{C}} \geqslant R + 2 \tag{4.26}$$

or

$$k'_{\mathbf{B}} = R \quad \textit{and} \quad k'_{\mathbf{A}} + k_{\mathbf{C}} \geqslant R + 2, \tag{4.27}$$

then $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ *and* $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$ *are essentially equal.*

Proof. It suffices to prove the theorem for condition (4.26). The result for (4.27) is obtained by switching modes.

From (2.5) we have

$$\mathbf{T}_{JK\times I} = (\mathbf{B} \odot \mathbf{C}) \cdot \mathbf{A}^T = (\bar{\mathbf{B}} \odot \bar{\mathbf{C}}) \cdot \bar{\mathbf{A}}^T. \tag{4.28}$$

From Lemma 4.3 we have

$$\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi}_a \cdot \mathbf{\Lambda}_a. \tag{4.29}$$

Since $k'_{\mathbf{A}} = R$, $\mathbf{A}$ is full column rank. Substitution of (4.29) in (4.28) now yields

$$\mathbf{B} \odot \mathbf{C} = (\bar{\mathbf{B}} \odot \bar{\mathbf{C}}) \cdot \mathbf{\Lambda}_a^T \cdot \mathbf{\Pi}_a^T. \tag{4.30}$$

This implies that the matrices $\bar{\mathbf{B}}_r \otimes \bar{\mathbf{c}}_r$ are ordered in the same way as the matrices $\bar{\mathbf{A}}_r$. Furthermore, if $\bar{\mathbf{A}}_i = \mathbf{A}_j \cdot \mathbf{L}$, with $\mathbf{L}$ nonsingular, then $\bar{\mathbf{B}}_i \otimes \bar{\mathbf{c}}_i = (\mathbf{B}_j \otimes \mathbf{c}_j) \cdot \mathbf{L}^{-T}$, or, equivalently, $\bar{\mathbf{B}}_i \circ \bar{\mathbf{c}}_i = (\mathbf{B}_j \cdot \mathbf{L}^{-T}) \circ \mathbf{c}_j$. □

We now prove that under a generalized Kruskal condition, the submatrices of $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ in an alternative decomposition of $\mathcal{T}$ are ordered in the same way.

LEMMA 4.6. *Let $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ represent a decomposition of $\mathcal{T}$ into rank-$(L_r, L_r, 1)$ terms, $1 \leqslant r \leqslant R$. Suppose that we have an alternative decomposition of $\mathcal{T}$, represented by $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$, with $k'_{\bar{\mathbf{A}}}$ and $k'_{\bar{\mathbf{B}}}$ maximal under the given dimensionality constraints. If the condition*

$$k'_{\mathbf{A}} + k'_{\mathbf{B}} + k_{\mathbf{C}} \geqslant 2R + 2 \tag{4.31}$$

holds, then $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}_a$ and $\bar{\mathbf{B}} = \mathbf{B} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}_b$, in which $\mathbf{\Pi}$ is a block permutation matrix and $\mathbf{\Lambda}_a$ and $\mathbf{\Lambda}_b$ nonsingular block-diagonal matrices, compatible with the block structure of $\mathbf{A}$ and $\mathbf{B}$.

Proof. From Lemma 4.3 we know that $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi}_a \cdot \mathbf{\Lambda}_a$ and $\bar{\mathbf{B}} = \mathbf{B} \cdot \mathbf{\Pi}_b \cdot \mathbf{\Lambda}_b$. We show that $\mathbf{\Pi}_a = \mathbf{\Pi}_b$ if (4.31) holds. We work in analogy with [38, pp. 129–132] and [54].

From (2.9) we have

$$\begin{aligned}\mathbf{T}_{I \times J, k} &= \mathbf{A} \cdot \operatorname{blockdiag}(c_{k1}\mathbf{I}_{L_1 \times L_1}, \ldots, c_{kR}\mathbf{I}_{L_R \times L_R}) \cdot \mathbf{B}^T \\ &= \bar{\mathbf{A}} \cdot \operatorname{blockdiag}(\bar{c}_{k1}\mathbf{I}_{L_1 \times L_1}, \ldots, \bar{c}_{kR}\mathbf{I}_{L_R \times L_R}) \cdot \bar{\mathbf{B}}^T.\end{aligned}$$

For vectors $\mathbf{v}$ and $\mathbf{w}$ we have

$$\begin{aligned}&(\mathbf{v}^T\mathbf{A}) \cdot \operatorname{blockdiag}(c_{k1}\mathbf{I}_{L_1 \times L_1}, \ldots, c_{kR}\mathbf{I}_{L_R \times L_R}) \cdot (\mathbf{w}^T\mathbf{B})^T \\ &\quad = (\mathbf{v}^T\bar{\mathbf{A}}) \cdot \operatorname{blockdiag}(\bar{c}_{k1}\mathbf{I}_{L_1 \times L_1}, \ldots, \bar{c}_{kR}\mathbf{I}_{L_R \times L_R}) \cdot (\mathbf{w}^T\bar{\mathbf{B}})^T \\ &\quad = (\mathbf{v}^T\mathbf{A}\mathbf{\Pi}_a) \cdot \mathbf{\Lambda}_a \cdot \operatorname{blockdiag}(\bar{c}_{k1}\mathbf{I}_{L_1 \times L_1}, \ldots, \bar{c}_{kR}\mathbf{I}_{L_R \times L_R}) \cdot \mathbf{\Lambda}_b^T \cdot (\mathbf{w}^T\mathbf{B}\mathbf{\Pi}_b)^T.\end{aligned} \tag{4.32}$$

We stack (4.32), for $k = 1, \ldots, K$, in

$$\begin{aligned}&\mathbf{C} \cdot \operatorname{blockdiag}(\mathbf{v}^T\mathbf{A}) \cdot \operatorname{blockdiag}(\mathbf{B}^T\mathbf{w}) \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \\ &= \bar{\mathbf{C}} \cdot \operatorname{blockdiag}(\mathbf{v}^T\mathbf{A}\mathbf{\Pi}_a) \cdot \mathbf{\Lambda}_a \cdot \mathbf{\Lambda}_b^T \cdot \operatorname{blockdiag}(\mathbf{\Pi}_b^T\mathbf{B}^T\mathbf{w}) \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.\end{aligned} \tag{4.33}$$

We define

$$\mathbf{p} = \operatorname{blockdiag}(\mathbf{v}^T\mathbf{A}) \cdot \operatorname{blockdiag}(\mathbf{B}^T\mathbf{w}) \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{v}^T\mathbf{A}_1 \cdot \mathbf{B}_1^T\mathbf{w} \\ \vdots \\ \mathbf{v}^T\mathbf{A}_R \cdot \mathbf{B}_R^T\mathbf{w} \end{pmatrix}.$$

Let the index function $g(x)$ be given by $\mathbf{A}\mathbf{\Pi}_a = \begin{pmatrix}\mathbf{A}_{g(1)} & \mathbf{A}_{g(2)} & \dots & \mathbf{A}_{g(R)}\end{pmatrix}$. Let a second index function $h(x)$ be given by $\mathbf{B}\mathbf{\Pi}_b = \begin{pmatrix}\mathbf{B}_{h(1)} & \mathbf{B}_{h(2)} & \dots & \mathbf{B}_{h(R)}\end{pmatrix}$. We define

$$\mathbf{q} = \text{blockdiag}(\mathbf{v}^T\mathbf{A}\mathbf{\Pi}_a)\cdot\mathbf{\Lambda}_a\cdot\mathbf{\Lambda}_b^T\cdot\text{blockdiag}(\mathbf{\Pi}_b^T\mathbf{B}^T\mathbf{w})\cdot\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}$$
$$= \begin{pmatrix}\mathbf{v}^T\mathbf{A}_{g(1)}\cdot\mathbf{\Lambda}_{a,1}\cdot\mathbf{\Lambda}_{b,1}^T\cdot\mathbf{B}_{h(1)}^T\mathbf{w}\\ \vdots\\ \mathbf{v}^T\mathbf{A}_{g(R)}\cdot\mathbf{\Lambda}_{a,R}\cdot\mathbf{\Lambda}_{b,R}^T\cdot\mathbf{B}_{h(R)}^T\mathbf{w}\end{pmatrix},$$

where $\mathbf{\Lambda}_{a,r}$ and $\mathbf{\Lambda}_{b,r}$ denote the rth block of $\mathbf{\Lambda}_a$ and $\mathbf{\Lambda}_b$, respectively.

Equation (4.33) can now be written as $\mathbf{C}\cdot\mathbf{p} = \bar{\mathbf{C}}\cdot\mathbf{q}$. Below we show by contradiction that $\mathbf{\Pi}_a = \mathbf{\Pi}_b$ if (4.31) holds. If $\mathbf{\Pi}_a \neq \mathbf{\Pi}_b$, then we will be able to find vectors $\mathbf{v}$ and $\mathbf{w}$ such that $\mathbf{q} = \mathbf{0}$ and $\mathbf{p} \neq \mathbf{0}$ has less than $k_{\mathbf{C}}$ nonzero elements. This implies that a set of less than $k_{\mathbf{C}}$ columns of $\mathbf{C}$ is linearly dependent, which contradicts the definition of $k_{\mathbf{C}}$.

Suppose that $\mathbf{\Pi}_a \neq \mathbf{\Pi}_b$. Then there exists an r such that $\mathbf{A}_r$ is the sth submatrix of $\mathbf{A}\mathbf{\Pi}_a$, $\mathbf{B}_r$ is the tth submatrix of $\mathbf{B}\mathbf{\Pi}_b$, and $s \neq t$. Formally, there exists an r such that $r = g(s) = h(t)$ and $s \neq t$. We now create two index sets $\mathsf{S}, \mathsf{T} \subset \{1,\dots,R\}$ as follows:

- Put $g(t)$ in S and $h(s)$ in T.
- For $x \in \{1,\dots,R\}\setminus\{s,t\}$, add $g(x)$ to S if $\text{card}(\mathsf{S}) < k'_{\mathbf{A}} - 1$. Otherwise, add $h(x)$ to T.

The sets S and T have the following properties. Since $k'_{\mathbf{A}} - 1 \leqslant R-1$, S contains exactly $k'_{\mathbf{A}} - 1$ elements. The set T contains $R - \text{card}(\mathsf{S}) = R - k'_{\mathbf{A}} + 1$ elements. Because of (4.31) and $k_{\mathbf{C}} \leqslant R$, this is less than or equal to $k'_{\mathbf{B}} - 1$ elements. In the xth element of $\mathbf{q}$ we have either $g(x) \in \mathsf{S}$ or $h(x) \in \mathsf{T}$, $x = 1,\dots,R$. The index $r = g(s) = h(t)$ is neither an element of S nor an element of T. Denote $\{i_1, i_2, \dots, i_{k'_{\mathbf{A}}-1}\} = \mathsf{S}$ and $\{j_1, j_2, \dots, j_{R-k'_{\mathbf{A}}+1}\} = \mathsf{T}$.

We choose vectors $\mathbf{v}$ and $\mathbf{w}$ such that $\mathbf{v}^T\mathbf{A}_i = \mathbf{0}$ if $i \in \mathsf{S}$, $\mathbf{w}^T\mathbf{B}_j = \mathbf{0}$ if $j \in \mathsf{T}$ and $\mathbf{v}^T\mathbf{A}_r\mathbf{B}_r^T\mathbf{w} \neq 0$. This is possible for the following reasons. By the definition of $k'_{\mathbf{A}}$, $[\mathbf{A}_{i_1} \ \dots \ \mathbf{A}_{i_{k'_{\mathbf{A}}-1}} \ \mathbf{A}_r]$ is full column rank. We have to choose $\mathbf{v}$ in $\text{null}([\mathbf{A}_{i_1} \ \dots \ \mathbf{A}_{i_{k'_{\mathbf{A}}-1}}])$. The projection of this subspace on $\text{span}(\mathbf{A}_r)$ is of dimension L_r. By varying $\mathbf{v}$ in $\text{null}([\mathbf{A}_{i_1} \ \dots \ \mathbf{A}_{i_{k'_{\mathbf{A}}-1}}])$, $\mathbf{v}^T\mathbf{A}_r$ can be made equal to any vector in $\mathbb{K}^{1\times L_r}$. For instance, we can choose $\mathbf{v}$ such that $\mathbf{v}^T\mathbf{A}_r = (1\ 0\ \dots\ 0)$. Similarly, we can choose a vector $\mathbf{w}$ in $\text{null}([\mathbf{B}_{j_1} \ \dots \ \mathbf{B}_{j_{R-k'_{\mathbf{A}}+1}}])$ satisfying $\mathbf{w}^T\mathbf{B}_r = (1\ 0\ \dots\ 0)$.

For the vectors $\mathbf{v}$ and $\mathbf{w}$ above, we have $\mathbf{q} = \mathbf{0}$. On the other hand, the rth element of $\mathbf{p}$ is nonzero. Define $\mathsf{S}^c = \{1,\dots,R\}\setminus\mathsf{S}$ and $\mathsf{T}^c = \{1,\dots,R\}\setminus\mathsf{T}$. The number of nonzero entries of $\mathbf{p}$ is bounded from above by

$$\text{card}(\mathsf{S}^c \cap \mathsf{T}^c) \leqslant \text{card}(\mathsf{S}^c) \leqslant R - k'_{\mathbf{A}} + 1 \leqslant k_{\mathbf{C}} - 1,$$

where the last inequality is due to (4.31) and $k'_{\mathbf{B}} \leqslant R$. Hence, $\mathbf{C}\cdot\mathbf{p} = \mathbf{0}$ implies that a set of less than $k_{\mathbf{C}}$ columns of $\mathbf{C}$ is linearly dependent, which contradicts the definition of $k_{\mathbf{C}}$. This completes the proof. □

THEOREM 4.7. *Let $(\mathbf{A},\mathbf{B},\mathbf{C})$ represent a decomposition of $\mathcal{T}$ in generic rank-$(L_r,L_r,1)$ terms, $1 \leqslant r \leqslant R$. Suppose that we have an alternative decomposition of $\mathcal{T}$, represented by $(\bar{\mathbf{A}},\bar{\mathbf{B}},\bar{\mathbf{C}})$, with $k'_{\bar{\mathbf{A}}}$ and $k'_{\bar{\mathbf{B}}}$ maximal under the given dimensionality*

constraints. If the conditions

$$IJ \geqslant \sum_{r=1}^{R} L_r^2, \tag{4.34}$$

$$k'_{\mathbf{A}} + k'_{\mathbf{B}} + k_{\mathbf{C}} \geqslant 2R + 2 \tag{4.35}$$

hold, then $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ *and* $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$ *are essentially equal.*

Proof. From Lemma 4.6 we have that $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}_a$ and $\bar{\mathbf{B}} = \mathbf{B} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}_b$. Put the submatrices of $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ in the same order as the submatrices of $\mathbf{A}$ and $\mathbf{B}$. After reordering, we have $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Lambda}_a$, with $\mathbf{\Lambda}_a = \text{blockdiag}(\mathbf{\Lambda}_{a,1}, \ldots, \mathbf{\Lambda}_{a,R})$, and $\bar{\mathbf{B}} = \mathbf{B} \cdot \mathbf{\Lambda}_b$, with $\mathbf{\Lambda}_b = \text{blockdiag}(\mathbf{\Lambda}_{b,1}, \ldots, \mathbf{\Lambda}_{b,R})$. From (2.4) we have that

$$\begin{aligned}
\mathbf{T}_{IJ\times K} &= (\mathbf{A} \odot_c \mathbf{B}) \cdot \text{blockdiag}(\mathbf{1}_{L_1}, \ldots, \mathbf{1}_{L_R}) \cdot \mathbf{C}^T \\
&= (\mathbf{A} \odot \mathbf{B}) \cdot \text{blockdiag}(\text{vec}(\mathbf{I}_{L_1 \times L_1}), \ldots, \text{vec}(\mathbf{I}_{L_R \times L_R})) \cdot \mathbf{C}^T \\
&= (\bar{\mathbf{A}} \odot \bar{\mathbf{B}}) \cdot \text{blockdiag}(\text{vec}(\mathbf{I}_{L_1 \times L_1}), \ldots, \text{vec}(\mathbf{I}_{L_R \times L_R})) \cdot \bar{\mathbf{C}}^T \\
&= (\mathbf{A} \odot \mathbf{B}) \cdot \text{blockdiag}(\text{vec}(\mathbf{\Lambda}_{a,1} \cdot \mathbf{\Lambda}_{b,1}^T), \ldots, \text{vec}(\mathbf{\Lambda}_{a,R} \cdot \mathbf{\Lambda}_{b,R}^T)) \cdot \bar{\mathbf{C}}^T. \qquad (4.36)
\end{aligned}$$

From [19, Lemma 3.3] we have that, under condition (4.34), $\mathbf{A} \odot \mathbf{B}$ is generically full column rank. Equation (4.36) then implies that there exist nonzero scalars α_r such that $\mathbf{\Lambda}_{a,r} \cdot \mathbf{\Lambda}_{b,r}^T = \alpha_r \mathbf{I}_{L_r \times L_r}$ (i.e., $\mathbf{\Lambda}_{a,r} = \alpha_r \mathbf{\Lambda}_{b,r}^{-T}$) and $\mathbf{c}_r = \alpha_r \bar{\mathbf{c}}_r$, $1 \leqslant r \leqslant R$. In other words, $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$ are equal up to trivial indeterminacies. ∎

5. The decomposition in rank-(L, M, N) terms. In this section we study the uniqueness of the decomposition in rank-(L, M, N) terms. We use the notation introduced in section 2.2. Section 5.1 concerns uniqueness of the general decomposition. In section 5.2 we have a closer look at the special case of rank-$(2, 2, 2)$ terms.

5.1. General results. In this section we follow the same structure as in section 4:

Theorem 5.1	corresponds to	Theorem 4.1
Lemma 5.2		Lemma 4.3
Theorem 5.3		Theorem 4.5
Lemma 5.4		Lemma 4.6
Theorem 5.5		Theorem 4.7.

For decompositions in generic rank-(L, M, N) terms, the results of this section can be summarized as follows. We have essential uniqueness if

(i) Theorem 5.1:

$$\begin{aligned}
&L = M \quad \text{and} \quad I \geqslant LR \quad \text{and} \quad J \geqslant MR \quad \text{and} \quad N \geqslant 3 \\
&\qquad \text{and} \quad \mathbf{C}_r \text{ is full column rank, } 1 \leqslant r \leqslant R; \qquad (5.1)
\end{aligned}$$

(ii) Theorem 5.3:

(5.2)

$$I \geqslant LR \quad \text{and} \quad N > L + M - 2 \quad \text{and} \quad \min\left(\left\lfloor \frac{J}{M} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{K}{N} \right\rfloor, R\right) \geqslant R + 2;$$

or

$$J \geqslant MR \quad \text{and} \quad N > L + M - 2 \quad \text{and} \quad \min\left(\left\lfloor \frac{I}{L} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{K}{N} \right\rfloor, R\right) \geqslant R + 2.$$

(5.3)

(iii) Theorem 5.5:

$$N > L + M - 2 \quad \text{and} \quad \min\left(\left\lfloor \frac{I}{L} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{J}{M} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{K}{N} \right\rfloor, R\right) \geqslant 2R + 2. \tag{5.4}$$

First we have a uniqueness result that stems from the fact that the column spaces of $\mathbf{A}_r$, $1 \leqslant r \leqslant R$, are invariant subspaces of quotients of tensor slices.

THEOREM 5.1. *Let* $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathcal{D})$ *represent a decomposition of* $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ *in* R *rank-*(L, L, N) *terms. Suppose that* $\operatorname{rank}(\mathbf{A}) = LR$, $\operatorname{rank}(\mathbf{B}) = LR$, $\operatorname{rank}_{k'}(\mathbf{C}) \geqslant 1$, $N \geqslant 3$, *and that* $\mathcal{D}$ *is generic. Then* $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathcal{D})$ *is essentially unique.*

Proof. From Theorem 6.1 below we have that under the conditions specified in Theorem 5.1, a decomposition in terms of the form $\mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r$ is essentially unique. Consequently, a decomposition in terms of the form $\mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r \bullet_3 \mathbf{C}$ is essentially unique if $\mathbf{C}$ is full column rank. A fortiori, reasoning as in the proof of Theorem 6.1, a decomposition in terms of the form $\mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r \bullet_3 \mathbf{C}_r$, in which the matrices $\mathbf{C}_r$ are possibly different, is essentially unique if these matrices $\mathbf{C}_r$ are full column rank. □

Remark 4. The generalization to the decomposition in rank-(L_r, L_r, N_r) terms, $1 \leqslant r \leqslant R$, is trivial.

Remark 5. In the nongeneric case, lack of uniqueness can be due to the fact that tensors $\mathcal{D}_r$ can be further block-diagonalized by means of basis transformations in their mode-1, mode-2, and mode-3 vector space. We give an example.

Example 1. Assume a tensor $\mathcal{T} \in \mathbb{K}^{12 \times 12 \times 12}$ that can be decomposed in three rank-$(4, 4, 4)$ terms as follows:

$$\mathcal{T} = \sum_{r=1}^{3} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r \bullet_3 \mathbf{C}_r$$

with $\mathcal{D}_r \in \mathbb{K}^{4 \times 4 \times 4}$, $\mathbf{A}_r, \mathbf{B}_r, \mathbf{C}_r \in \mathbb{K}^{12 \times 4}$, $1 \leqslant r \leqslant 3$. Now assume that $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ can be further decomposed as follows:

$$\begin{aligned}
\mathcal{D}_1 &= \mathbf{u}_1 \circ \mathbf{v}_1 \circ \mathbf{w}_1 + \mathbf{u}_2 \circ \mathbf{v}_2 \circ \mathbf{w}_2 + \mathcal{H}_1 \bullet_1 \mathbf{E}_1 \bullet_2 \mathbf{F}_1 \bullet_3 \mathbf{G}_1, \\
\mathcal{D}_2 &= \mathbf{u}_3 \circ \mathbf{v}_3 \circ \mathbf{w}_3 + \mathcal{H}_2 \bullet_1 \mathbf{E}_2 \bullet_2 \mathbf{F}_2 \bullet_3 \mathbf{G}_2, \\
\mathcal{D}_3 &= \mathbf{u}_4 \circ \mathbf{v}_4 \circ \mathbf{w}_4 + \mathcal{H}_3 \bullet_1 \mathbf{E}_3 \bullet_2 \mathbf{F}_3 \bullet_3 \mathbf{G}_3,
\end{aligned}$$

where $\mathbf{u}_s, \mathbf{v}_s, \mathbf{w}_s \in \mathbb{K}^4$, $1 \leqslant s \leqslant 4$, $\mathbf{E}_1, \mathbf{F}_1, \mathbf{G}_1 \in \mathbb{K}^{4 \times 2}$, $\mathbf{E}_2, \mathbf{E}_3, \mathbf{F}_2, \mathbf{F}_3, \mathbf{G}_2, \mathbf{G}_3 \in \mathbb{K}^{4 \times 3}$, $\mathcal{H}_1 \in \mathbb{K}^{2 \times 2 \times 2}$, $\mathcal{H}_2, \mathcal{H}_3 \in \mathbb{K}^{3 \times 3 \times 3}$. Then we have the following alternative decomposition in three rank-$(4, 4, 4)$ terms:

$$\begin{aligned}
\mathcal{T} = {} & [(\mathbf{A}_2\mathbf{u}_3) \circ (\mathbf{B}_2\mathbf{v}_3) \circ (\mathbf{C}_2\mathbf{w}_3) + (\mathbf{A}_3\mathbf{u}_4) \circ (\mathbf{B}_3\mathbf{v}_4) \circ (\mathbf{C}_3\mathbf{w}_4) \\
& + \mathcal{H}_1 \bullet_1 (\mathbf{A}_1\mathbf{E}_1) \bullet_2 (\mathbf{B}_1\mathbf{F}_1) \bullet_3 (\mathbf{C}_1\mathbf{G}_1)] \\
& + [(\mathbf{A}_1\mathbf{u}_1) \circ (\mathbf{B}_1\mathbf{v}_1) \circ (\mathbf{C}_1\mathbf{w}_1) + \mathcal{H}_2 \bullet_1 (\mathbf{A}_2\mathbf{E}_2) \bullet_2 (\mathbf{B}_2\mathbf{F}_2) \bullet_3 (\mathbf{C}_2\mathbf{G}_2)] \\
& + [(\mathbf{A}_1\mathbf{u}_2) \circ (\mathbf{B}_1\mathbf{v}_2) \circ (\mathbf{C}_1\mathbf{w}_2) + \mathcal{H}_3 \bullet_1 (\mathbf{A}_3\mathbf{E}_3) \bullet_2 (\mathbf{B}_3\mathbf{F}_3) \bullet_3 (\mathbf{C}_3\mathbf{G}_3)].
\end{aligned}$$

We now prove essential uniqueness of $\mathbf{A}$ and $\mathbf{B}$ under a constraint on the block dimensions and a Kruskal-type condition.

LEMMA 5.2. *Let* $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathcal{D})$ *represent a decomposition of* $\mathcal{T}$ *in* R *rank-*(L, M, N) *terms. Suppose that the conditions*

$$N > L + M - 2, \tag{5.5}$$

$$k'_{\mathbf{A}} + k'_{\mathbf{B}} + k'_{\mathbf{C}} \geqslant 2R + 2 \tag{5.6}$$

hold and that we have an alternative decomposition of $\mathcal{T}$*, represented by* $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathcal{D}})$*, with* $k'_{\bar{\mathbf{A}}}$ *and* $k'_{\bar{\mathbf{B}}}$ *maximal under the given dimensionality constraints. For generic* $\mathcal{D}$ *there holds that* $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi}_a \cdot \mathbf{\Lambda}_a$*, in which* $\mathbf{\Pi}_a$ *is a block permutation matrix and* $\mathbf{\Lambda}_a$ *a square nonsingular block-diagonal matrix, compatible with the structure of* $\mathbf{A}$*. There also holds* $\bar{\mathbf{B}} = \mathbf{B} \cdot \mathbf{\Pi}_b \cdot \mathbf{\Lambda}_b$*, in which* $\mathbf{\Pi}_b$ *is a block permutation matrix and* $\mathbf{\Lambda}_b$ *a square nonsingular block-diagonal matrix, compatible with the structure of* $\mathbf{B}$*.*

Proof. It suffices to prove the lemma for $\mathbf{A}$. The result for $\mathbf{B}$ can be obtained by switching modes. We work in analogy with [54] and the proof of Lemma 4.2 and 4.3. We use the equivalence lemma for partitioned matrices to prove essential uniqueness of $\mathbf{A}$.

(i) *Derivation of an upper-bound on* $\omega'(\mathbf{x}^T \bar{\mathbf{A}})$. The constraint on $k'_{\bar{\mathbf{A}}}$ implies that $k'_{\bar{\mathbf{A}}} \geqslant k'_{\mathbf{A}}$. Hence, if $\omega'(\mathbf{x}^T \bar{\mathbf{A}}) \leqslant R - k'_{\bar{\mathbf{A}}} + 1$, then

$$\omega'(\mathbf{x}^T \bar{\mathbf{A}}) \leqslant R - k'_{\bar{\mathbf{A}}} + 1 \leqslant R - k'_{\mathbf{A}} + 1 \leqslant k'_{\mathbf{B}} + k_{\mathbf{C}} - (R + 1), \tag{5.7}$$

where the last inequality corresponds to condition (5.6).

(ii) *Derivation of a lower-bound on* $\omega'(\mathbf{x}^T \bar{\mathbf{A}})$. Consider $\mathcal{D}_r \bullet_1 (\mathbf{x}^T \mathbf{A}_r)$ and $\bar{\mathcal{D}}_r \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_r)$, $1 \leqslant r \leqslant R$, as $(M \times N)$ matrices. Then the linear combination of slices $\sum_{i=1}^{I} x_i \mathbf{T}_{J \times K, i}$ is given by

$$\begin{aligned} &\mathbf{B} \cdot \text{blockdiag}[\mathcal{D}_1 \bullet_1 (\mathbf{x}^T \mathbf{A}_1), \ldots, \mathcal{D}_R \bullet_1 (\mathbf{x}^T \mathbf{A}_R)] \cdot \mathbf{C}^T \\ &\quad = \bar{\mathbf{B}} \cdot \text{blockdiag}[\bar{\mathcal{D}}_1 \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_1), \ldots, \bar{\mathcal{D}}_R \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_R)] \cdot \bar{\mathbf{C}}^T. \end{aligned}$$

Taking into account that $N > M$, we have

$$\begin{aligned} M\omega'(\mathbf{x}^T \bar{\mathbf{A}}) &\geqslant r_{\text{blockdiag}[\bar{\mathcal{D}}_1 \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_1), \ldots, \bar{\mathcal{D}}_R \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_R)]} \\ &\geqslant r_{\bar{\mathbf{B}} \cdot \text{blockdiag}[\bar{\mathcal{D}}_1 \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_1), \ldots, \bar{\mathcal{D}}_R \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_R)] \cdot \bar{\mathbf{C}}^T} \\ &= r_{\mathbf{B} \cdot \text{blockdiag}[\mathcal{D}_1 \bullet_1 (\mathbf{x}^T \mathbf{A}_1), \ldots, \mathcal{D}_R \bullet_1 (\mathbf{x}^T \mathbf{A}_R)] \cdot \mathbf{C}^T}. \end{aligned} \tag{5.8}$$

Since the tensors $\mathcal{D}_r$ are generic, and because of condition (5.5), all the $(M \times N)$ matrices $\mathcal{D}_r \bullet_1 (\mathbf{x}^T \mathbf{A}_r)$ are rank-M. (Rank deficiency would imply that $N - M + 1$ determinants are zero, while $\mathbf{x}$ provides only $L - 1$ independent parameters and an irrelevant scaling factor.) Define $(K \times M)$ matrices $\underline{\mathbf{C}}_r = \mathbf{C}_r \cdot [\mathcal{D}_r \bullet_1 (\mathbf{x}^T \mathbf{A}_r)]^T$, $1 \leqslant r \leqslant R$. Let $\gamma = \omega'(\mathbf{x}^T \mathbf{A})$ and $\underline{\mathbf{C}} = (\underline{\mathbf{C}}_1 \ \ldots \ \underline{\mathbf{C}}_R)$. Let $\tilde{\mathbf{B}}$ and $\tilde{\underline{\mathbf{C}}}$ consist of the submatrices of $\mathbf{B}$ and $\underline{\mathbf{C}}$, respectively, corresponding to the parts of $\mathbf{x}^T \mathbf{A}$ that are not all-zero. From (5.8) we have

$$M\omega'(\mathbf{x}^T \bar{\mathbf{A}}) \geqslant r_{\tilde{\mathbf{B}} \cdot \tilde{\underline{\mathbf{C}}}^T}. \tag{5.9}$$

Both $\tilde{\mathbf{B}}$ and $\tilde{\underline{\mathbf{C}}}$ have γM columns. Sylvester's inequality now yields

$$r_{\tilde{\mathbf{B}} \cdot \tilde{\underline{\mathbf{C}}}^T} \geqslant r_{\tilde{\mathbf{B}}} + r_{\tilde{\underline{\mathbf{C}}}} - \gamma M. \tag{5.10}$$

From the definition of k′-rank, we have

$$r_{\tilde{\mathbf{B}}} \geqslant M \min(\gamma, k'_{\mathbf{B}}). \tag{5.11}$$

On the other hand, $\underline{\tilde{\mathbf{C}}}$ consists of γ $(K \times M)$ submatrices, of which the columns are sampled in the column space of the corresponding submatrix of $\mathbf{C}$. From the definition of k′-rank, we must have

$$r_{\underline{\tilde{\mathbf{C}}}} \geqslant M \min(\gamma, k'_{\mathbf{C}}). \tag{5.12}$$

Combination of (5.9)–(5.12) yields the following lower-bound on $\omega'(\mathbf{x}^T \bar{\mathbf{A}})$:

$$\omega'(\mathbf{x}^T \bar{\mathbf{A}}) \geqslant \min(\gamma, k'_{\mathbf{B}}) + \min(\gamma, k'_{\mathbf{C}}) - \gamma. \tag{5.13}$$

(iii) *Combination of the two bounds.* This is analogous to Lemma 4.2. □

If matrix $\mathbf{A}$ or $\mathbf{B}$ is tall and full column rank, then its essential uniqueness implies essential uniqueness of the overall tensor decomposition.

THEOREM 5.3. *Let* $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathcal{D})$ *represent a decomposition of* $\mathcal{T}$ *in* R *rank-*(L, M, N) *terms, with* $N > L + M - 2$. *Suppose that we have an alternative decomposition of* $\mathcal{T}$, *represented by* $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathcal{D}})$, *with* $k'_{\bar{\mathbf{A}}}$ *and* $k'_{\bar{\mathbf{B}}}$ *maximal under the given dimensionality constraints. For generic* $\mathcal{D}$ *there holds that if*

$$k'_{\mathbf{A}} = R \quad \textit{and} \quad k'_{\mathbf{B}} + k'_{\mathbf{C}} \geqslant R + 2 \tag{5.14}$$

or

$$k'_{\mathbf{B}} = R \quad \textit{and} \quad k'_{\mathbf{A}} + k'_{\mathbf{C}} \geqslant R + 2, \tag{5.15}$$

then $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathcal{D})$ *and* $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathcal{D}})$ *are essentially equal.*

Proof. It suffices to prove the theorem for $\mathbf{A}$. The result for $\mathbf{B}$ is obtained by switching modes. From (2.12) we have

$$\begin{aligned} \mathbf{T}_{JK \times I} &= (\mathbf{B} \odot \mathbf{C}) \cdot \text{blockdiag}((\mathcal{D}_1)_{MN \times L}, \ldots, (\mathcal{D}_R)_{MN \times L}) \cdot \mathbf{A}^T \\ &= (\bar{\mathbf{B}} \odot \bar{\mathbf{C}}) \cdot \text{blockdiag}((\bar{\mathcal{D}}_1)_{MN \times L}, \ldots, (\bar{\mathcal{D}}_R)_{MN \times L}) \cdot \bar{\mathbf{A}}^T. \end{aligned} \tag{5.16}$$

From Lemma 5.2 we have

$$\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi}_a \cdot \mathbf{\Lambda}_a. \tag{5.17}$$

Since $k'_{\mathbf{A}} = R$, $\mathbf{A}$ is full column rank. Substitution of (5.17) in (5.16) now yields

$$\begin{aligned} &(\mathbf{B} \odot \mathbf{C}) \cdot \text{blockdiag}((\mathcal{D}_1)_{MN \times L}, \ldots, (\mathcal{D}_R)_{MN \times L}) \\ &\quad = (\bar{\mathbf{B}} \odot \bar{\mathbf{C}}) \cdot \text{blockdiag}((\bar{\mathcal{D}}_1)_{MN \times L}, \ldots, (\bar{\mathcal{D}}_R)_{MN \times L}) \cdot \mathbf{\Lambda}_a^T \cdot \mathbf{\Pi}_a^T. \end{aligned} \tag{5.18}$$

This implies that the matrices $(\bar{\mathbf{B}}_r \otimes \bar{\mathbf{C}}_r) \cdot (\bar{\mathcal{D}}_r)_{MN \times L}$ are permuted in the same way with respect to $(\mathbf{B}_r \otimes \mathbf{C}_r) \cdot (\mathcal{D}_r)_{MN \times L}$ as the matrices $\bar{\mathbf{A}}_r$ with respect to $\mathbf{A}_r$. Furthermore, if $\bar{\mathbf{A}}_i = \mathbf{A}_j \cdot \mathbf{F}$, then $(\bar{\mathbf{B}}_i \otimes \bar{\mathbf{C}}_i) \cdot (\bar{\mathcal{D}}_i)_{MN \times L} \cdot \mathbf{F}^T = (\mathbf{B}_j \otimes \mathbf{C}_j) \cdot (\mathcal{D}_j)_{MN \times L}$. Equivalently, we have $\bar{\mathcal{D}}_i \bullet_2 \bar{\mathbf{B}}_i \bullet_3 \bar{\mathbf{C}}_i = \mathcal{D}_j \bullet_1 \mathbf{F}^{-1} \bullet_2 \mathbf{B}_j \bullet_3 \mathbf{C}_j$. □

We now prove that under conditions (5.5) and (5.6), the submatrices of $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ in an alternative decomposition of $\mathcal{T}$ are ordered in the same way.

LEMMA 5.4. *Let* $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathcal{D})$ *represent a decomposition of* $\mathcal{T}$ *in* R *rank-*(L, M, N) *terms. Suppose that we have an alternative decomposition of* $\mathcal{T}$, *represented by* $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathcal{D}})$, *with* $k'_{\bar{\mathbf{A}}}$ *and* $k'_{\bar{\mathbf{B}}}$ *maximal under the given dimensionality constraints. For generic* $\mathcal{D}$ *there holds that if conditions* (5.5) *and* (5.6) *hold, then* $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}_a$ *and* $\bar{\mathbf{B}} = \mathbf{B} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}_b$, *in which* $\mathbf{\Pi}$ *is a block permutation matrix and* $\mathbf{\Lambda}_a$ *and* $\mathbf{\Lambda}_b$ *square nonsingular block-diagonal matrices, compatible with the block structure of* $\mathbf{A}$ *and* $\mathbf{B}$.

Proof. From Lemma 5.2 we know that $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi}_a \cdot \mathbf{\Lambda}_a$ and $\bar{\mathbf{B}} = \mathbf{B} \cdot \mathbf{\Pi}_b \cdot \mathbf{\Lambda}_b$. We show that $\mathbf{\Pi}_a = \mathbf{\Pi}_b$ if (5.5) and (5.6) hold. We work in analogy with [38, pp. 129–132], [54], and the proof of Lemma 4.6.

Since both $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathcal{D})$ and $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathcal{D}})$ represent a decomposition of $\mathcal{T}$, we have for vectors $\mathbf{v}$ and $\mathbf{w}$,

$$\begin{aligned} \mathcal{T} \bullet_1 \mathbf{v}^T \bullet_2 \mathbf{w}^T &= \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 (\mathbf{v}^T \mathbf{A}_r) \bullet_2 (\mathbf{w}^T \mathbf{B}_r) \bullet_3 \mathbf{C}_r \\ &= \sum_{r=1}^{R} \bar{\mathcal{D}}_r \bullet_1 (\mathbf{v}^T \bar{\mathbf{A}}_r) \bullet_2 (\mathbf{w}^T \bar{\mathbf{B}}_r) \bullet_3 \bar{\mathbf{C}}_r. \end{aligned} \tag{5.19}$$

Let the index functions $g(x)$ and $h(x)$ be given by $\mathbf{A\Pi}_a = \begin{pmatrix} \mathbf{A}_{g(1)} & \mathbf{A}_{g(2)} & \dots & \mathbf{A}_{g(R)} \end{pmatrix}$ and $\mathbf{B\Pi}_b = \begin{pmatrix} \mathbf{B}_{h(1)} & \mathbf{B}_{h(2)} & \dots & \mathbf{B}_{h(R)} \end{pmatrix}$, respectively. Then (5.19) can be written as

$$\mathbf{C} \cdot \mathbf{p} = \bar{\mathbf{C}} \cdot \mathbf{q}, \tag{5.20}$$

in which $\mathbf{p}$ and $\mathbf{q}$ are defined by

$$\mathbf{p} = \begin{pmatrix} (\mathcal{D}_1)_{N \times LM} \cdot [(\mathbf{A}_1^T \mathbf{v}) \otimes (\mathbf{B}_1^T \mathbf{w})] \\ \vdots \\ (\mathcal{D}_R)_{N \times LM} \cdot [(\mathbf{A}_R^T \mathbf{v}) \otimes (\mathbf{B}_R^T \mathbf{w})] \end{pmatrix},$$

$$\mathbf{q} = \begin{pmatrix} (\bar{\mathcal{D}}_1)_{N \times LM} \cdot [(\mathbf{\Lambda}_{a,1}^T \bar{\mathbf{A}}_{g(1)}^T \mathbf{v}) \otimes (\mathbf{\Lambda}_{b,1}^T \bar{\mathbf{B}}_{h(1)}^T \mathbf{w})] \\ \vdots \\ (\bar{\mathcal{D}}_R)_{N \times LM} \cdot [(\mathbf{\Lambda}_{a,R}^T \bar{\mathbf{A}}_{g(R)}^T \mathbf{v}) \otimes (\mathbf{\Lambda}_{b,R}^T \bar{\mathbf{B}}_{h(R)}^T \mathbf{w})] \end{pmatrix},$$

where $\mathbf{\Lambda}_{a,r}$ and $\mathbf{\Lambda}_{b,r}$ denote the rth block of $\mathbf{\Lambda}_a$ and $\mathbf{\Lambda}_b$, respectively.

We will now show by contradiction that $\mathbf{\Pi}_a = \mathbf{\Pi}_b$. If $\mathbf{\Pi}_a \neq \mathbf{\Pi}_b$, then we will be able to find vectors $\mathbf{v}$ and $\mathbf{w}$ such that $\mathbf{q} = \mathbf{0}$ and $\mathbf{p} \neq \mathbf{0}$ has less than $k'_{\mathbf{C}}$ nonzero $(N \times 1)$ subvectors. This implies that a set of less than $k'_{\mathbf{C}}$ vectors, each sampled in the column space of a different submatrix of $\mathbf{C}$, is linearly dependent, which contradicts the definition of $k'_{\mathbf{C}}$.

Suppose that $\mathbf{\Pi}_a \neq \mathbf{\Pi}_b$. Then there exists an r such that $\mathbf{A}_r$ is the sth submatrix of $\mathbf{A\Pi}_a$, $\mathbf{B}_r$ is the tth submatrix of $\mathbf{B\Pi}_b$, and $s \neq t$. Formally, there exists an r such that $r = g(s) = h(t)$ and $s \neq t$. We now create two index sets $\mathsf{S}, \mathsf{T} \subset \{1, \dots, R\}$ in the same way as in the proof of Lemma 4.6.

Since $k'_{\mathbf{A}} - 1 \leqslant R - 1$, S contains exactly $k'_{\mathbf{A}} - 1$ elements. The set T contains $R - \text{card}(\mathsf{S}) = R - k'_{\mathbf{A}} + 1$ elements. Because of (5.6) and $k'_{\mathbf{C}} \leqslant R$, this is less than or equal to $k'_{\mathbf{B}} - 1$ elements. In the xth element of $\mathbf{q}$ we have either $g(x) \in \mathsf{S}$ or $h(x) \in \mathsf{T}$, $x = 1, \dots, R$. The index $r = g(s) = h(t)$ is neither an element of S nor an element of T. Denote $\{i_1, i_2, \dots, i_{k'_{\mathbf{A}}-1}\} = \mathsf{S}$ and $\{j_1, j_2, \dots, j_{R-k'_{\mathbf{A}}+1}\} = \mathsf{T}$.

We choose a vector $\mathbf{v}$ such that $\mathbf{v}^T \mathbf{A}_i = \mathbf{0}$ if $i \in \mathsf{S}$, and $\mathbf{v}^T \mathbf{A}_r \neq 0$. This is always possible. The vector $\mathbf{v}$ has to be chosen in $\text{null}([\mathbf{A}_{i_1} \ \dots \ \mathbf{A}_{i_{k'_{\mathbf{A}}-1}}])$, which is an $(I - (k'_{\mathbf{A}} - 1)L)$-dimensional space. If a column of $\mathbf{A}_r$ is orthogonal to all possible vectors $\mathbf{v}$, then it lies in $\text{span}([\mathbf{A}_{i_1} \ \dots \ \mathbf{A}_{i_{k'_{\mathbf{A}}-1}}])$. Then we would have a contradiction with the definition of $k'_{\mathbf{A}}$. Similarly, we can choose a vector $\mathbf{w}$ such that $\mathbf{w}^T \mathbf{B}_j = \mathbf{0}$ if $j \in \mathsf{T}$, and $\mathbf{w}^T \mathbf{B}_r \neq 0$.

Because of condition (5.5), the genericity of $\mathcal{D}_r$, and the fact that $\mathbf{v}^T \mathbf{A}_r \neq 0$, the $(N \times M)$ matrix $\mathcal{D}_r \bullet_1 (\mathbf{v}^T \mathbf{A}_r)$ is rank-M. Rank deficiency would imply that $N - M + 1$

determinants are zero, while $\mathbf{v}^T\mathbf{A}_r$ provides only $L-1$ parameters and an irrelevant scaling factor. Since $\mathcal{D}_r \bullet_1 (\mathbf{v}^T\mathbf{A}_r)$ is full column rank, and since $\mathbf{w}^T\mathbf{B}_r \neq 0$, we have $\mathcal{D}_r \bullet_1 (\mathbf{v}^T\mathbf{A}_r) \bullet_2 (\mathbf{w}^T\mathbf{B}_r) \neq \mathbf{0}$. Equivalently, $(\mathcal{D}_r)_{N\times LM} \cdot [(\mathbf{A}_r^T\mathbf{v}) \otimes (\mathbf{B}_r^T\mathbf{w})] \neq \mathbf{0}$.

Define $\mathsf{S}^c = \{1,\ldots,R\} \setminus \mathsf{S}$ and $\mathsf{T}^c = \{1,\ldots,R\} \setminus \mathsf{T}$. The number of nonzero subvectors of $\mathbf{p}$ is bounded from above by

$$\mathrm{card}(\mathsf{S}^c \cap \mathsf{T}^c) \leqslant \mathrm{card}(\mathsf{S}^c) \leqslant R - k'_{\mathbf{A}} + 1 \leqslant k'_{\mathbf{C}} - 1,$$

where the last inequality is due to (5.6) and $k'_{\mathbf{B}} \leqslant R$. Hence, $\mathbf{C}\cdot\mathbf{p} = \mathbf{0}$ implies that a set of less than $k'_{\mathbf{C}}$ columns, each sampled in the column space of a different submatrix of $\mathbf{C}$, is linearly dependent, which contradicts the definition of $k'_{\mathbf{C}}$. This completes the proof. ∎

THEOREM 5.5. *Let $(\mathbf{A},\mathbf{B},\mathbf{C},\mathcal{D})$ represent a decomposition of $\mathcal{T}$ in R rank-(L,M,N) terms. Suppose that we have an alternative decomposition of $\mathcal{T}$, represented by $(\bar{\mathbf{A}},\bar{\mathbf{B}},\bar{\mathbf{C}},\bar{\mathcal{D}})$, with $k'_{\bar{\mathbf{A}}}$ and $k'_{\bar{\mathbf{B}}}$ maximal under the given dimensionality constraints. For generic $\mathcal{D}$ there holds that, if conditions* (5.5) *and* (5.6) *hold, then $(\mathbf{A},\mathbf{B},\mathbf{C},\mathcal{D})$ and $(\bar{\mathbf{A}},\bar{\mathbf{B}},\bar{\mathbf{C}},\bar{\mathcal{D}})$ are essentially equal.*

Proof. From Lemma 5.4 we have that $\bar{\mathbf{A}} = \mathbf{A}\cdot\mathbf{\Pi}\cdot\mathbf{\Lambda}_a$ and $\bar{\mathbf{B}} = \mathbf{B}\cdot\mathbf{\Pi}\cdot\mathbf{\Lambda}_b$. Put the submatrices of $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ in the same order as the submatrices of $\mathbf{A}$ and $\mathbf{B}$. After reordering, we have $\bar{\mathbf{A}} = \mathbf{A}\cdot\mathbf{\Lambda}_a$, with $\mathbf{\Lambda}_a = \mathrm{blockdiag}(\mathbf{\Lambda}_{a,1},\ldots,\mathbf{\Lambda}_{a,R})$, and $\bar{\mathbf{B}} = \mathbf{B}\cdot\mathbf{\Lambda}_b$, with $\mathbf{\Lambda}_b = \mathrm{blockdiag}(\mathbf{\Lambda}_{b,1},\ldots,\mathbf{\Lambda}_{b,R})$. From (2.14) we have that

$$\begin{aligned}\mathbf{T}_{IJ\times K} &= (\mathbf{A}\odot\mathbf{B})\cdot\mathrm{blockdiag}((\mathcal{D}_1)_{LM\times N},\ldots,(\mathcal{D}_R)_{LM\times N})\cdot\mathbf{C}^T\\ &= (\bar{\mathbf{A}}\odot\bar{\mathbf{B}})\cdot\mathrm{blockdiag}((\bar{\mathcal{D}}_1)_{LM\times N},\ldots,(\bar{\mathcal{D}}_R)_{LM\times N})\cdot\bar{\mathbf{C}}^T\\ &= (\mathbf{A}\odot\mathbf{B})\cdot\mathrm{blockdiag}((\mathbf{\Lambda}_{a,1}\otimes\mathbf{\Lambda}_{b,1})\cdot(\bar{\mathcal{D}}_1)_{LM\times N},\ldots,\\ &\qquad (\mathbf{\Lambda}_{a,R}\otimes\mathbf{\Lambda}_{b,R})\cdot(\bar{\mathcal{D}}_R)_{LM\times N})\cdot\bar{\mathbf{C}}^T. \qquad (5.21)\end{aligned}$$

From Lemma 3.4 we have that $k'_{\mathbf{A}\odot\mathbf{B}} \geqslant \min(k'_{\mathbf{A}} + k'_{\mathbf{B}} - 1, R)$. From (5.6) we have that $k'_{\mathbf{A}} + k'_{\mathbf{B}} - 1 \geqslant 2R + 1 - k'_{\mathbf{C}} \geqslant R+1$. Hence, $k'_{\mathbf{A}\odot\mathbf{B}} = R$, which implies that $\mathbf{A}\odot\mathbf{B}$ is full column rank. Multiplying (5.21) from the left by $(\mathbf{A}\odot\mathbf{B})^\dagger$, we obtain that

$$(\mathbf{\Lambda}_{a,r}\otimes\mathbf{\Lambda}_{b,r})\cdot(\bar{\mathcal{D}}_r)_{LM\times N}\cdot\bar{\mathbf{C}}_r^T = (\mathcal{D}_r)_{LM\times N}\cdot\mathbf{C}_r^T, \quad 1\leqslant r\leqslant R.$$

This can be rewritten as

$$\bar{\mathcal{D}}_r \bullet_1 \mathbf{\Lambda}_{a,r} \bullet_2 \mathbf{\Lambda}_{b,r} \bullet_3 \bar{\mathbf{C}}_r = \mathcal{D}_r \bullet_3 \mathbf{C}_r, \quad 1\leqslant r\leqslant R.$$

This means that $(\mathbf{A},\mathbf{B},\mathbf{C},\mathcal{D})$ and $(\bar{\mathbf{A}},\bar{\mathbf{B}},\bar{\mathbf{C}},\bar{\mathcal{D}})$ are equal up to trivial indeterminacies. ∎

5.2. Rank-(2, 2, 2) blocks. In the Kruskal-type results of the previous section, we have only considered rank-(L,M,N) terms for which $N > L+M-2$. Rank-$(2,2,3)$ terms, for instance, satisfy this condition. However, it would also be interesting to know whether the decomposition of a tensor in rank-$(2,2,2)$ terms is essentially unique. This special case is studied in this section.

A first result is that in $\mathbb{C}$ the decomposition of a tensor $\mathcal{T}$ in $R\geqslant 2$ rank-$(2,2,2)$ terms is not essentially unique. This is easy to understand. Assume, for instance, that $\mathcal{T}$ is the sum of two rank-$(2,2,2)$ terms $\mathcal{T}_1$ and $\mathcal{T}_2$. It is well known that in $\mathbb{C}$ the rank of rank-$(2,2,2)$ tensor is always equal to 2 [55]. Hence we have for some vectors $\mathbf{a}_r$,

$\mathbf{b}_r$, $\mathbf{c}_r$, $1 \leqslant r \leqslant 4$,

$$\begin{aligned}\mathcal{T} &= \mathcal{T}_1 + \mathcal{T}_2 \\ &= (\mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1 + \mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2) + (\mathbf{a}_3 \circ \mathbf{b}_3 \circ \mathbf{c}_3 + \mathbf{a}_4 \circ \mathbf{b}_4 \circ \mathbf{c}_4) \\ &= (\mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1 + \mathbf{a}_3 \circ \mathbf{b}_3 \circ \mathbf{c}_3) + (\mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2 + \mathbf{a}_4 \circ \mathbf{b}_4 \circ \mathbf{c}_4) \\ &= \tilde{\mathcal{T}}_1 + \tilde{\mathcal{T}}_2.\end{aligned}$$

Since $\tilde{\mathcal{T}}_1$ and $\tilde{\mathcal{T}}_2$ yield an other decomposition, the decomposition of $\mathcal{T}$ in 2 rank-$(2,2,2)$ terms is not essentially unique.

Theorem 5.5 does not hold in the case of rank-$(2,2,2)$ terms because Lemma 5.2 does not hold. The problem is that in (5.8) the (2×2) matrices $\mathcal{D}_r \times_1 (\mathbf{x}^T \mathbf{A}_r)$ are not necessarily rank-2. Indeed, let λ be a generalized eigenvalue of the pencil formed by the (2×2) matrices $(\mathcal{D}_r)_{1,:,:}$ and $(\mathcal{D}_r)_{2,:,:}$. Then $\mathcal{D}_r \bullet_1 (\mathbf{x}^T \mathbf{A}_r)$ is rank-1 if $\mathbf{x}^T \mathbf{A}_r$ is proportional to $(1, -\lambda)$. As a result, (5.12) does not hold.

On the other hand, if we work in $\mathbb{R}$, the situation is somewhat different. In $\mathbb{R}$, rank-$(2,2,2)$ terms can be either rank-2 or rank-3 [30, 39, 55]. If $\mathcal{D}_r$ is rank-2 in $\mathbb{R}$, then the pencil $((\mathcal{D}_r)_{1,:,:}, (\mathcal{D}_r)_{2,:,:})$ has two real generalized eigenvalues. Conversely, if the generalized eigenvalues of $((\mathcal{D}_r)_{1,:,:}, (\mathcal{D}_r)_{2,:,:})$ are complex, then $\mathcal{D}_r$ is rank-3. (The tensor $\mathcal{D}_r$ can also be rank-3 when an eigenvalue has algebraic multiplicity two but geometric multiplicity one. This case occurs with probability zero when the entries of $\mathcal{D}_r$ are drawn from continuous probability density functions and will not further be considered in this section.) We now have the following variant of Theorem 5.5.

THEOREM 5.6. *Let* $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathcal{D})$ *represent a real decomposition of* $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$ *in* R *rank-*$(2,2,2)$ *terms. Suppose that the condition*

$$k'_{\mathbf{A}} + k'_{\mathbf{B}} + k'_{\mathbf{C}} \geqslant 2R + 2$$

holds and that the generalized eigenvalues of $((\mathcal{D}_r)_{1,:,:}, (\mathcal{D}_r)_{2,:,:})$ *are complex,* $1 \leqslant r \leqslant R$. *Then* $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathcal{D})$ *is essentially unique.*

Proof. Under the condition on the generalized eigenvalues of $((\mathcal{D}_r)_{1,:,:}, (\mathcal{D}_r)_{2,:,:})$, the matrices $\mathcal{D}_r \bullet_1 (\mathbf{x}^T \mathbf{A}_r)$ in (5.8) are necessarily rank-2, and the reasoning in the proof of Lemma 5.2 remains valid.

On the other hand, assuming that $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi}_a \cdot \mathbf{\Lambda}_a$ and $\bar{\mathbf{B}} = \mathbf{B} \cdot \mathbf{\Pi}_b \cdot \mathbf{\Lambda}_b$, only a technical modification of the proof of Lemma 5.4 is required to make sure that $\mathbf{\Pi}_a = \mathbf{\Pi}_b$ does hold. We only have to verify whether vectors $\mathbf{v}$ and $\mathbf{w}$ can be found such that $\mathbf{v}^T \mathbf{A}_i = \mathbf{0}$ if $i \in \mathsf{S}$, $\mathbf{w}^T \mathbf{B}_j = \mathbf{0}$ if $j \in \mathsf{T}$, and $(\mathcal{D}_r)_{N \times LM} \cdot [(\mathbf{A}_r^T \mathbf{v}) \otimes (\mathbf{B}_r^T \mathbf{w})] \neq \mathbf{0}$. Reasoning as in the proof of Lemma 5.4, we see that the constraint $\mathbf{v}^T \mathbf{A}_i = \mathbf{0}$, $i \in \mathsf{S}$, still leaves enough freedom for $\mathbf{v}^T \mathbf{A}_r$ to be any vector in $\mathbb{R}^2$. Equivalently, the constraint $\mathbf{w}^T \mathbf{B}_j = \mathbf{0}$, $j \in \mathsf{T}$, leaves enough freedom for $\mathbf{w}^T \mathbf{B}_r$ to be any vector in $\mathbb{R}^2$. We conclude that it is always possible to find the required vectors $\mathbf{v}$ and $\mathbf{w}$ if $\mathcal{D}_r \neq \mathcal{O}$.

Essential uniqueness of the overall tensor decomposition now follows from $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}_a$ and $\bar{\mathbf{B}} = \mathbf{B} \cdot \mathbf{\Pi} \cdot \mathbf{\Lambda}_b$ in the same way as in the proof of Theorem 5.5. ∎

From Theorem 5.6 follows that a generic decomposition in real rank-3 rank-$(2,2,2)$ terms is essentially unique provided,

$$\min\left(\left\lfloor \frac{I}{2} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{J}{2} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{K}{2} \right\rfloor, R\right) \geqslant 2R + 2.$$

Finally, we have the following variant of Theorem 5.1.

THEOREM 5.7. *Let* $(\mathbf{A},\mathbf{B},\mathbf{C},\mathcal{D})$ *represent a real decomposition of* $\mathcal{T}\in\mathbb{R}^{I\times J\times K}$ *in* R *rank-*(L,M,N) *terms, with* $L=M=N=2$. *Suppose that* $\operatorname{rank}(\mathbf{A})=2R$, $\operatorname{rank}(\mathbf{B})=2R$, $\operatorname{rank}_{k'}(\mathbf{C})\geqslant 1$ *and that all generalized eigenvalues of the pencil* $((\mathcal{D}_r)_{L\times M,1},(\mathcal{D}_r)_{L\times M,2})$ *are complex,* $1\leqslant r\leqslant R$. *Then* $(\mathbf{A},\mathbf{B},\mathbf{C},\mathcal{D})$ *is essentially unique.*

Proof. Consider two vectors $\mathbf{x},\mathbf{y}\in\mathbb{R}^K$ for which $\mathbf{x}^T\mathbf{C}_r$ is not proportional to $\mathbf{y}^T\mathbf{C}_r$, $1\leqslant r\leqslant R$. Since all matrices $\mathbf{C}_r$ are full column rank, this is the case for generic vectors $\mathbf{x},\mathbf{y}$. Define $\mathbf{T}_1=\sum_{k=1}^K x_k\mathbf{T}_{I\times J,k}$ and $\mathbf{T}_2=\sum_{k=1}^K y_k\mathbf{T}_{I\times J,k}$. We have

$$\begin{aligned}\mathbf{T}_2\cdot\mathbf{T}_1^\dagger=\mathbf{A}\cdot\operatorname{blockdiag}\{&([\mathcal{D}_1\bullet_3(\mathbf{y}^T\mathbf{C}_1)]\cdot[\mathcal{D}_1\bullet_3(\mathbf{x}^T\mathbf{C}_1)]^\dagger,\ldots,\\&[\mathcal{D}_R\bullet_3(\mathbf{y}^T\mathbf{C}_R)]\cdot[\mathcal{D}_R\bullet_3(\mathbf{x}^T\mathbf{C}_R)]^\dagger)\}\cdot\mathbf{A}^\dagger.\end{aligned}$$

From this equation it is clear that the column space of any $\mathbf{A}_r$ is an invariant subspace of $\mathbf{T}_2\cdot\mathbf{T}_1^\dagger$.

Define $\mathbf{C}_r^T\mathbf{x}=\tilde{\mathbf{x}}_r$ and $\mathbf{C}_r^T\mathbf{y}=\tilde{\mathbf{y}}_r$. We have

$$\begin{aligned}\mathcal{D}_r\bullet_3(\mathbf{x}^T\mathbf{C}_r)&=(\tilde{\mathbf{x}}_r)_1(\mathcal{D}_r)_{L\times M,1}+(\tilde{\mathbf{x}}_r)_2(\mathcal{D}_r)_{L\times M,2},\\\mathcal{D}_r\bullet_3(\mathbf{y}^T\mathbf{C}_r)&=(\tilde{\mathbf{y}}_r)_1(\mathcal{D}_r)_{L\times M,1}+(\tilde{\mathbf{y}}_r)_2(\mathcal{D}_r)_{L\times M,2}.\end{aligned}$$

If there exist real values α and β, with $\alpha^2+\beta^2=1$, such that $\alpha\,\mathcal{D}_r\bullet_3(\mathbf{x}^T\mathbf{C}_r)+\beta\,\mathcal{D}_r\bullet_3(\mathbf{y}^T\mathbf{C}_r)$ is rank-1, then there also exist real values γ en μ, with $\gamma^2+\mu^2=1$, such that $\gamma\,(\mathcal{D}_r)_{L\times M,1}+\mu\,(\mathcal{D}_r)_{L\times M,2}$ is rank-1. The condition on the generalized eigenvalues of the pencils $((\mathcal{D}_r)_{L\times M,1},(\mathcal{D}_r)_{L\times M,2})$ implies thus that the blocks $[\mathcal{D}_r\bullet_3(\mathbf{y}^T\mathbf{C}_r)]\cdot[\mathcal{D}_r\bullet_3(\mathbf{x}^T\mathbf{C}_r)]^\dagger$ cannot be diagonalized by means of a real similarity transformation. We conclude that the only two-dimensional invariant subspaces of $\mathbf{T}_2\cdot\mathbf{T}_1^\dagger$ are the column spaces of the matrices $\mathbf{A}_r$. In other words, $\mathbf{A}$ is essentially unique.

Essential uniqueness of the overall decomposition now follows from (2.12). Assume that we have an alternative decomposition of $\mathcal{T}$, represented by $(\bar{\mathbf{A}},\bar{\mathbf{B}},\bar{\mathbf{C}},\bar{\mathcal{D}})$. We have $\bar{\mathbf{A}}=\mathbf{A}\cdot\mathbf{\Pi}_a\cdot\mathbf{\Lambda}_a$, in which $\mathbf{\Pi}_a$ is a block-permutation matrix and $\mathbf{\Lambda}_a=\operatorname{blockdiag}(\mathbf{\Lambda}_{a,1},\ldots,\mathbf{\Lambda}_{a,R})$ a square nonsingular block-diagonal matrix, compatible with the block structure of $\mathbf{A}$. From (2.12) we have

$$\begin{aligned}\mathbf{T}_{JK\times I}&=(\mathbf{B}\odot\mathbf{C})\cdot\operatorname{blockdiag}((\mathcal{D}_1)_{MN\times L},\ldots,(\mathcal{D}_R)_{MN\times L})\cdot\mathbf{A}^T\\&=(\bar{\mathbf{B}}\odot\bar{\mathbf{C}})\cdot\operatorname{blockdiag}((\bar{\mathcal{D}}_1)_{MN\times L},\ldots,(\bar{\mathcal{D}}_R)_{MN\times L})\cdot\mathbf{\Pi}_a^T\cdot\mathbf{\Lambda}_a^T\cdot\mathbf{A}^T.\end{aligned}$$

Right multiplication by $(\mathbf{A}^T)^\dagger$ yields

$$\begin{aligned}&(\mathbf{B}\odot\mathbf{C})\cdot\operatorname{blockdiag}((\mathcal{D}_1)_{MN\times L},\ldots,(\mathcal{D}_R)_{MN\times L})\\&=(\bar{\mathbf{B}}\odot\bar{\mathbf{C}})\cdot\operatorname{blockdiag}((\bar{\mathcal{D}}_1)_{MN\times L},\ldots,(\bar{\mathcal{D}}_R)_{MN\times L})\cdot\mathbf{\Pi}_a^T\cdot\mathbf{\Lambda}_a^T.\end{aligned}\tag{5.22}$$

Assume that the rth submatrix of $\mathbf{A}$ corresponds to the s-th submatrix of $\bar{\mathbf{A}}$. Then we have from (5.22) that

$$(\mathbf{B}_r\odot\mathbf{C}_r)\cdot(\mathcal{D}_r)_{MN\times L}=(\bar{\mathbf{B}}_s\odot\bar{\mathbf{C}}_s)\cdot(\bar{\mathcal{D}}_s)_{MN\times L}\cdot\mathbf{\Lambda}_{a,s}^T$$

in which $\mathbf{\Lambda}_{a,s}$ is the sth block of $\mathbf{\Lambda}_a$. Equivalently,

$$\mathcal{D}_r\bullet_2\mathbf{B}_r\bullet_3\mathbf{C}_r=\bar{\mathcal{D}}_s\bullet_1\mathbf{\Lambda}_{a,s}\bullet_2\bar{\mathbf{B}}_s\bullet_3\bar{\mathbf{C}}_s.$$

This completes the proof. ∎

6. Type-2 decomposition in rank-$(L, M, \cdot)$ terms. In this section we derive several conditions under which the type-2 decomposition in rank-$(L, M, \cdot)$ terms is unique. We use the notation introduced in section 2.3.

First we have a uniqueness result that stems from the fact that the column spaces of $\mathbf{A}_r$, $1 \leqslant r \leqslant R$, are invariant subspaces of quotients of tensor slices. This result is the counterpart of Theorem 4.1 in section 4 and Theorem 5.1 in section 5.1.

THEOREM 6.1. *Let* $(\mathbf{A}, \mathbf{B}, \mathcal{C})$ *represent a type-*2 *decomposition of* $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ *in* R *rank-*$(L, L, \cdot)$ *terms. Suppose that* $\operatorname{rank}(\mathbf{A}) = LR$, $\operatorname{rank}(\mathbf{B}) = LR$, $K \geqslant 3$, *and that* $\mathcal{C}$ *is generic. Then* $(\mathbf{A}, \mathbf{B}, \mathcal{C})$ *is essentially unique.*

Proof. We have

$$\mathbf{T}_{I\times J,2} \cdot \mathbf{T}^{\dagger}_{I\times J,1} = \mathbf{A} \cdot \operatorname{blockdiag}((\mathcal{C}_1)_{L\times M,2} \cdot (\mathcal{C}_1)^{\dagger}_{L\times M,1}, \ldots, (\mathcal{C}_R)_{L\times M,2} \cdot (\mathcal{C}_R)^{\dagger}_{L\times M,1}) \cdot \mathbf{A}^{\dagger},$$

where $M = L$. From this equation it is clear that the column space of any $\mathbf{A}_r$ is an invariant subspace of $\mathbf{T}_{I\times J,2} \cdot \mathbf{T}^{\dagger}_{I\times J,1}$. However, any set of eigenvectors forms an invariant subspace. To determine which eigenvectors belong together, we use the third slice $\mathbf{T}_{I\times J,3}$. We have

(6.1)
$$\mathbf{T}_{I\times J,3} \cdot \mathbf{T}^{\dagger}_{I\times J,1} = \mathbf{A} \cdot \operatorname{blockdiag}((\mathcal{C}_1)_{L\times M,3} \cdot (\mathcal{C}_1)^{\dagger}_{L\times M,1}, \ldots, (\mathcal{C}_R)_{L\times M,3} \cdot (\mathcal{C}_R)^{\dagger}_{L\times M,1}) \cdot \mathbf{A}^{\dagger}.$$

It is clear that the column space of any $\mathbf{A}_r$ is also an invariant subspace of $\mathbf{T}_{I\times J,3} \cdot \mathbf{T}^{\dagger}_{I\times J,1}$. On the other hand, because of the genericity of $\mathcal{C}$, we can interpret $(\mathcal{C}_r)_{L\times M,3} \cdot (\mathcal{C}_r)^{\dagger}_{L\times M,1}$ as $(\mathcal{C}_r)_{L\times M,2} \cdot (\mathcal{C}_r)^{\dagger}_{L\times M,1} + \mathbf{E}_r$, in which $\mathbf{E}_r \in \mathbb{K}^{L\times L}$ is a generic perturbation, $1 \leqslant r \leqslant R$. Perturbation analysis now states that the individual eigenvectors of $\mathbf{T}_{I\times J,3} \cdot \mathbf{T}^{\dagger}_{I\times J,1}$ do not correspond to those of $\mathbf{T}_{I\times J,2} \cdot \mathbf{T}^{\dagger}_{I\times J,1}$ [23, 32]. We conclude that $\mathbf{A}$ is essentially unique.

Essential uniqueness of the overall decomposition follows directly from the essential uniqueness of $\mathbf{A}$. Assume that we have an alternative decomposition of $\mathcal{T}$, represented by $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathcal{C}})$. We have $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi}_a \cdot \mathbf{\Lambda}_a$, in which $\mathbf{\Pi}_a$ is a block-permutation matrix and $\mathbf{\Lambda}_a$ a square nonsingular block-diagonal matrix, compatible with the block structure of $\mathbf{A}$. From (2.18) we have

$$\begin{aligned}\mathbf{T}_{JK\times I} &= [(\mathcal{C}_1 \bullet_2 \mathbf{B}_1)_{JK\times L} \ \ldots \ (\mathcal{C}_R \bullet_2 \mathbf{B}_R)_{JK\times L}] \cdot \mathbf{A}^T \\ &= \left[(\bar{\mathcal{C}}_1 \bullet_2 \bar{\mathbf{B}}_1)_{JK\times L} \ \ldots \ (\bar{\mathcal{C}}_R \bullet_2 \bar{\mathbf{B}}_R)_{JK\times L}\right] \cdot \bar{\mathbf{A}}^T.\end{aligned}$$

Hence,

$$\begin{aligned}&[(\mathcal{C}_1 \bullet_2 \mathbf{B}_1)_{JK\times L} \ \ldots \ (\mathcal{C}_R \bullet_2 \mathbf{B}_R)_{JK\times L}] \\ &\quad = \left[(\bar{\mathcal{C}}_1 \bullet_2 \bar{\mathbf{B}}_1)_{JK\times L} \ \ldots \ (\bar{\mathcal{C}}_R \bullet_2 \bar{\mathbf{B}}_R)_{JK\times L}\right] \cdot \mathbf{\Lambda}_a^T \cdot \mathbf{\Pi}_a^T.\end{aligned}$$

This implies that the matrices $(\mathcal{C}_r \bullet_2 \mathbf{B}_r)_{JK\times L}$ are ordered in the same way as the matrices $\mathbf{A}_r$. Furthermore, if $\bar{\mathbf{A}}_i = \mathbf{A}_j \cdot \mathbf{F}$, then $(\bar{\mathcal{C}}_i \bullet_2 \bar{\mathbf{B}}_i)_{JK\times L} \cdot \mathbf{F}^T = (\mathcal{C}_j \bullet_2 \mathbf{B}_j)_{JK\times L}$. Equivalently, we have $\bar{\mathcal{C}}_i \bullet_2 \bar{\mathbf{B}}_i = \mathcal{C}_j \bullet_1 \mathbf{F}^{-1} \bullet_2 \mathbf{B}_j$. This means that $(\mathbf{A}, \mathbf{B}, \mathcal{C})$ and $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathcal{C}})$ are essentially equal. ∎

Remark 6. The generalization to the decomposition in rank-$(L_r, L_r, \cdot)$ terms, $1 \leqslant r \leqslant R$, is trivial.

Remark 7. In the nongeneric case, lack of uniqueness can be due to the fact that tensors $\mathcal{C}_r$ can be subdivided in smaller blocks by means of basis transformations in their mode-1 and mode-2 vector space. We give an example.

Example 2. Consider a tensor $\mathcal{T} \in \mathbb{K}^{10\times 10\times 5}$ that can be decomposed in two rank-$(5,5,\cdot)$ terms as follows:

$$\mathcal{T} = \sum_{r=1}^{2} \mathcal{C}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r$$

with $\mathcal{C}_r \in \mathbb{K}^{5\times 5\times 5}$, $\mathbf{A}_r \in \mathbb{K}^{10\times 5}$, and $\mathbf{B}_r \in \mathbb{K}^{10\times 5}$, $1 \leqslant r \leqslant 2$. Now assume that $\mathcal{C}_1$ and $\mathcal{C}_2$ can be further decomposed as follows:

$$\begin{aligned}\mathcal{C}_1 &= \mathcal{G}_{11} \bullet_1 \mathbf{E}_{11} \bullet_2 \mathbf{F}_{11} + \mathcal{G}_{12} \bullet_1 \mathbf{E}_{12} \bullet_2 \mathbf{F}_{12},\\ \mathcal{C}_2 &= \mathcal{G}_{21} \bullet_1 \mathbf{E}_{21} \bullet_2 \mathbf{F}_{21} + \mathcal{G}_{22} \bullet_1 \mathbf{E}_{22} \bullet_2 \mathbf{F}_{22},\end{aligned}$$

where $\mathcal{G}_{11}, \mathcal{G}_{21} \in \mathbb{K}^{2\times 2\times 5}$, $\mathcal{G}_{12}, \mathcal{G}_{22} \in \mathbb{K}^{3\times 3\times 5}$, $\mathbf{E}_{11}, \mathbf{E}_{21}, \mathbf{F}_{11}, \mathbf{F}_{21} \in \mathbb{K}^{5\times 2}$, $\mathbf{E}_{12}, \mathbf{E}_{22}$, $\mathbf{F}_{12}, \mathbf{F}_{22} \in \mathbb{K}^{5\times 3}$. Define

$$\begin{aligned}\tilde{\mathbf{A}}_1 &= [\mathbf{A}_1 \cdot \mathbf{E}_{11} \ \mathbf{A}_2 \cdot \mathbf{E}_{22}], \qquad \tilde{\mathbf{A}}_2 = [\mathbf{A}_2 \cdot \mathbf{E}_{21} \ \mathbf{A}_1 \cdot \mathbf{E}_{12}],\\ \tilde{\mathbf{B}}_1 &= [\mathbf{B}_1 \cdot \mathbf{F}_{11} \ \mathbf{B}_2 \cdot \mathbf{F}_{22}], \qquad \tilde{\mathbf{B}}_2 = [\mathbf{B}_2 \cdot \mathbf{F}_{21} \ \mathbf{B}_1 \cdot \mathbf{F}_{12}],\end{aligned}$$

$$\begin{aligned}&(\tilde{\mathcal{C}}_1)_{1:2,1:2,:} = \mathcal{G}_{11}, \quad (\tilde{\mathcal{C}}_1)_{3:5,3:5,:} = \mathcal{G}_{22}, \quad (\tilde{\mathcal{C}}_1)_{1:2,3:5,:} = \mathcal{O}, \quad (\tilde{\mathcal{C}}_1)_{3:5,1:2,:} = \mathcal{O},\\ &(\tilde{\mathcal{C}}_2)_{1:2,1:2,:} = \mathcal{G}_{21}, \quad (\tilde{\mathcal{C}}_2)_{3:5,3:5,:} = \mathcal{G}_{12}, \quad (\tilde{\mathcal{C}}_2)_{1:2,3:5,:} = \mathcal{O}, \quad (\tilde{\mathcal{C}}_2)_{3:5,1:2,:} = \mathcal{O}.\end{aligned}$$

Then an alternative decomposition of $\mathcal{T}$ in rank-$(5,5,\cdot)$ terms is given by

$$\mathcal{T} = \sum_{r=1}^{2} \tilde{\mathcal{C}}_r \bullet_1 \tilde{\mathbf{A}}_r \bullet_2 \tilde{\mathbf{B}}_r. \tag{6.2}$$

For the case in which $\mathcal{C}_r \in \mathbb{R}^{2\times 2\times 2}$, $1 \leqslant r \leqslant R$, we have the following theorem.

THEOREM 6.2. *Let* $(\mathbf{A}, \mathbf{B}, \mathcal{C})$ *represent a real type-2 decomposition of* $\mathcal{T} \in \mathbb{R}^{I\times J\times 2}$ *in* R *rank-*$(L, M, 2)$ *terms with* $L = M = 2$. *Suppose that* $\operatorname{rank}(\mathbf{A}) = 2R$, $\operatorname{rank}(\mathbf{B}) = 2R$ *and that all generalized eigenvalues of the pencil* $((\mathcal{C}_r)_{L\times M,1}, (\mathcal{C}_r)_{L\times M,2})$ *are complex,* $1 \leqslant r \leqslant R$. *Then* $(\mathbf{A}, \mathbf{B}, \mathcal{C})$ *is essentially unique.*

Proof. This theorem is a special case of Theorem 5.7. The tensors $\mathcal{D}_r$ in Theorem 5.7 correspond to $\mathcal{C}_r$, and the matrices $\mathbf{C}_r$ in Theorem 5.7 are equal to $\mathbf{I}_{2\times 2}$. ∎

In some cases, uniqueness of the decomposition can be demonstrated by direct application of the equivalence lemma for partitioned matrices. This is illustrated in the following example.

Example 3. We show that the decomposition of a tensor $\mathcal{T} \in \mathbb{K}^{5\times 6\times 6}$ in $R = 3$ generic rank-$(2,2,\cdot)$ terms is essentially unique. Denote $I = 5$, $J = K = 6$, and $L = M = 2$. Let the decomposition be represented by $(\mathbf{A}, \mathbf{B}, \mathcal{C})$ and let us assume the existence of an alternative decomposition, represented by $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathcal{C}})$, which is "nonsingular" in the sense that the columns of $\bar{\mathbf{A}}$ are as linearly independent as possible.

To show that $(\mathbf{A}, \mathbf{B}, \mathcal{C})$ and $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathcal{C}})$ are essentially equal, we first use the equivalence lemma for partitioned matrices to show that $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi}_a \cdot \mathbf{\Lambda}_a$, in which $\mathbf{\Pi}_a$ is a block permutation matrix and $\mathbf{\Lambda}_a$ a square nonsingular block-diagonal matrix, both consisting of (2×2) blocks. We show that for every $\mu \leqslant R - k'_{\bar{\mathbf{A}}} + 1 = 2$ there holds that for a generic vector $\mathbf{x} \in \mathbb{K}^5$ such that $\omega'(\mathbf{x}^T\bar{\mathbf{A}}) \leqslant \mu$, we have $\omega'(\mathbf{x}^T\mathbf{A}) \leqslant \omega'(\mathbf{x}^T\bar{\mathbf{A}})$. We will subsequently examine the different cases corresponding to $\mu = 0, 1, 2$.

We first derive an inequality that will prove useful. Denote by $(\mathbf{C}_r \bullet_1 (\mathbf{x}^T\mathbf{A}_r))_{M\times K}$ the $(M \times K)$ matrix formed by the single slice of $\mathbf{C}_r \bullet_1 (\mathbf{x}^T\mathbf{A}_r)$, and denote by

$(\bar{\mathbf{C}}_r \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_r))_{M\times K}$ the $(M \times K)$ matrix formed by the single slice of $\bar{\mathbf{C}}_r \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_r)$, $1 \leqslant r \leqslant R$. Then the $(J \times K)$ matrix formed by the single slice of $\mathcal{T} \bullet_1 \mathbf{x}^T$ is given by

$$\bar{\mathbf{B}} \cdot \begin{pmatrix} (\bar{\mathbf{C}}_1 \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_1))_{M\times K} \\ \vdots \\ (\bar{\mathbf{C}}_R \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_R))_{M\times K} \end{pmatrix} = \mathbf{B} \cdot \begin{pmatrix} (\mathbf{C}_1 \bullet_1 (\mathbf{x}^T \mathbf{A}_1))_{M\times K} \\ \vdots \\ (\mathbf{C}_R \bullet_1 (\mathbf{x}^T \mathbf{A}_R))_{M\times K} \end{pmatrix}.$$

For the rank of this matrix, we have

$$\begin{aligned} M\omega'(\mathbf{x}^T \bar{\mathbf{A}}) &\geqslant \operatorname{rank} \left[\bar{\mathbf{B}} \cdot \begin{pmatrix} (\bar{\mathbf{C}}_1 \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_1))_{M\times K} \\ \vdots \\ (\bar{\mathbf{C}}_R \bullet_1 (\mathbf{x}^T \bar{\mathbf{A}}_R))_{M\times K} \end{pmatrix} \right] \\ &= \operatorname{rank} \left[\mathbf{B} \cdot \begin{pmatrix} (\mathbf{C}_1 \bullet_1 (\mathbf{x}^T \mathbf{A}_1))_{M\times K} \\ \vdots \\ (\mathbf{C}_R \bullet_1 (\mathbf{x}^T \mathbf{A}_R))_{M\times K} \end{pmatrix} \right]. \end{aligned}$$

Let $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{D}}(\mathbf{x})^T$ consist of the submatrices of $\mathbf{B}$ and

$$\begin{pmatrix} (\mathbf{C}_1 \bullet_1 (\mathbf{x}^T \mathbf{A}_1))_{M\times K} \\ \vdots \\ (\mathbf{C}_R \bullet_1 (\mathbf{x}^T \mathbf{A}_R))_{M\times K} \end{pmatrix},$$

respectively, corresponding to the nonzero subvectors of $\mathbf{x}^T \mathbf{A}$. Then we have

$$M\omega'(\mathbf{x}^T \bar{\mathbf{A}}) \geqslant r_{\tilde{\mathbf{B}} \cdot \tilde{\mathbf{D}}(\mathbf{x})^T}.$$

Since $\mathbf{B}$ is generic, we have

$$M\omega'(\mathbf{x}^T \bar{\mathbf{A}}) \geqslant r_{\tilde{\mathbf{D}}(\mathbf{x})^T}. \tag{6.3}$$

First, note that due to the "nonsingularity" of $\bar{\mathbf{A}}$, there does not exist a vector $\mathbf{x}$ such that $\omega'(\mathbf{x}^T \bar{\mathbf{A}}) = 0$. This means that the case $\mu = 0$ does not present a difficulty.

Next, we consider the case $\mu = 1$. Since $\omega'(\mathbf{x}^T \bar{\mathbf{A}}) \leqslant \mu$, we have that $M\omega'(\mathbf{x}^T \bar{\mathbf{A}})$ in (6.3) is less than or equal to 2. Since $\mathbf{x}$ is orthogonal to two submatrices of $\bar{\mathbf{A}}$, the set V of vectors $\mathbf{x}$ satisfying $\omega'(\mathbf{x}^T \bar{\mathbf{A}}) \leqslant \mu$ is the union of three one-dimensional subspaces in $\mathbb{K}^5$. We prove by contradiction that for a generic $\mathbf{x} \in \mathsf{V}$, we have $\omega'(\mathbf{x}^T \mathbf{A}) \leqslant 1$. Assume first that $\omega'(\mathbf{x}^T \mathbf{A}) = 2$. Then $\tilde{\mathbf{D}}(\mathbf{x})$ in (6.3) is a (6×4) matrix. For this (6×4) matrix to be rank-2, eight independent conditions on $\mathbf{x}$ have to be satisfied. (This value is the total number of entries (i.e., 24) minus the number of independent parameters in a (6×4) rank-2 matrix (i.e., 16). The latter value can easily be determined as the number of independent parameters in, for instance, an SVD.) These conditions can impossibly be satisfied in a subset of V that is not of measure zero. We conclude that for a generic $\mathbf{x} \in \mathsf{V}$, $\omega'(\mathbf{x}^T \mathbf{A}) \neq 2$. Next assume that $\omega'(\mathbf{x}^T \mathbf{A}) = 3$. Then $\tilde{\mathbf{D}}(\mathbf{x})$ in (6.3) is a (6×6) matrix. For this matrix to be rank-2, $36 - 20 = 10$ independent conditions on $\mathbf{x}$ have to be satisfied. We conclude that for a generic $\mathbf{x}$, $\omega'(\mathbf{x}^T \mathbf{A}) \neq 3$. This completes the case $\mu = 1$.

Finally, we consider the case $\mu = 2$. We now have that $M\omega'(\mathbf{x}^T \bar{\mathbf{A}})$ in (6.3) is less than or equal to 4. Since $\mathbf{x}$ is orthogonal to one submatrix of $\bar{\mathbf{A}}$, the set V of vectors $\mathbf{x}$

satisfying $\omega'(\mathbf{x}^T\bar{\mathbf{A}}) \leqslant \mu$ is the union of three three-dimensional subspaces in $\mathbb{K}^5$. We prove by contradiction that for a generic $\mathbf{x} \in \mathsf{V}$, we have $\omega'(\mathbf{x}^T\mathbf{A}) \leqslant 2$. Assume that $\omega'(\mathbf{x}^T\mathbf{A}) = 3$. Then $\tilde{\mathbf{D}}(\mathbf{x})$ in (6.3) is a (6×6) matrix. For this matrix to be rank-4, $36 - 32 = 4$ independent conditions on $\mathbf{x}$ have to be satisfied. These conditions can impossibly be satisfied in a subset of V that is not of measure zero. This completes the case $\mu = 2$.

We conclude that the condition of the equivalence lemma for partitioned matrices is satisfied. Hence, $\bar{\mathbf{A}} = \mathbf{A} \cdot \mathbf{\Pi}_a \cdot \mathbf{\Lambda}_a$. Essential uniqueness of the decomposition follows directly from the essential uniqueness of $\mathbf{A}$; cf. the proof of Theorem 6.1.

7. Discussion and future research. In this paper we introduced the concept of block term decompositions. A block term decomposition of a tensor $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ decomposes the given $(I \times J \times K)$-dimensional block in a number of blocks of smaller size. The size of a block is characterized by its mode-n rank triplet. (We mean the following. Consider a rank-(L, M, N) tensor $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$. The observed dimensions of $\mathcal{T}$ are I, J, K. However, its inner dimensions, its inherent size, are given by L, M, N.) The number of blocks that are needed in a decomposition depends on the size of the blocks. On the other hand, the number of blocks that is allowed determines which size they should minimally be.

The concept of block term decompositions unifies HOSVD/Tucker's decomposition and CANDECOMP/PARAFAC. HOSVD is a meaningful representation of a rank-(L, M, N) tensor as a single block of size (L, M, N). PARAFAC decomposes a rank-R tensor in R scalar blocks.

In the case of matrices, column rank and row rank are equal; moreover, they are equal to the minimal number of rank-1 terms in which the matrix can be decomposed. This is a consequence of the fact that matrices can be diagonalized by means of basis transformations in their column and row space. On the other hand, tensors cannot in general be diagonalized by means of basis transformations in their mode-1, mode-2, and mode-3 vector space. This has led to the distinction between mode-n rank triplet and rank. Like HOSVD and PARAFAC, these are the two extrema in a spectrum. It is interesting to note that "the" rank of a higher-order tensor is actually a combination of the two aspects: one should specify the number of blocks *and* their size. This is not clear at the matrix level because of the lack of uniqueness of decompositions in nonscalar blocks.

Matrices can actually be diagonalized by means of orthogonal (unitary) basis transformations in their column and row space. On the other hand, by imposing orthogonality constraints on PARAFAC one obtains different (approximate) decompositions, with different properties [8, 35, 36, 42]. Generalizations to block decompositions can easily be formulated. For instance, the generalization of [8, 42] to decompositions in rank-(L, M, N) terms is simply obtained by claiming that $\mathbf{A}_r^H \cdot \mathbf{A}_s = \mathbf{0}_{L\times L}$, $\mathbf{B}_r^H \cdot \mathbf{B}_s = \mathbf{0}_{M\times M}$, and $\mathbf{C}_r^H \cdot \mathbf{C}_s = \mathbf{0}_{N\times N}$, $1 \leqslant r \neq s \leqslant R$.

Interestingly enough, the generalization of different aspects of the matrix SVD most often leads to different tensor decompositions. Although the definition of block term decompositions is very general, tensor SVDs that do not belong to this class do exist. For instance, a variational definition of singular values and singular vectors was generalized in [41]. Although Tucker's decomposition and the best rank-(L, M, N) approximation can be obtained by means of a variational approach [13, 15, 61], the general theory does not fit in the framework of block decompositions.

Block term decompositions have an interesting interpretation in terms of the decomposition of homogeneous polynomials or multilinear forms. The PARAFAC de-

composition of a fully symmetric tensor (i.e., a tensor that is invariant under arbitrary index permutations) can be interpreted in terms of the decomposition of the associated homogeneous polynomial (quantic) in a sum of powers of linear forms [9]. For block term decompositions we now have the following. Given the quantic, linear forms are defined and clustered in subsets. Only within the same subset, products are admissible. The block term decomposition then decomposes the quantic in a sum of admissible products.

For instance, let $\mathcal{P} \in \mathbb{K}^{I\times I\times I}$ be fully symmetric. Let $\mathbf{x} \in \mathbb{K}^I$ be a vector of unknowns. Associate the quantic $p(\mathbf{x}) = \mathcal{P} \bullet_1 \mathbf{x}^T \bullet_2 \mathbf{x}^T \bullet_3 \mathbf{x}^T$ to $\mathcal{P}$. Let a PARAFAC decomposition of $\mathcal{P}$ be given by

$$\mathcal{P} = \sum_{r=1}^{R} d_r \, \mathbf{a}_r \circ \mathbf{a}_r \circ \mathbf{a}_r.$$

Define $y_r = \mathbf{x}^T\mathbf{a}_r$, $1 \leqslant r \leqslant R$. Then the quantic can be written as

$$p(\mathbf{y}) = \sum_{r=1}^{R} d_r \, y_r^3.$$

On the other hand, let a decomposition of $\mathcal{P}$ in rank-(L_r, L_r, L_r) terms be given by

$$\mathcal{P} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{A}_r \bullet_3 \mathbf{A}_r,$$

in which $\mathcal{D}_r \in \mathbb{K}^{L_r\times L_r\times L_r}$ and $\mathbf{A}_r \in \mathbb{K}^{I\times L_r}$, $1 \leqslant r \leqslant R$. Define $y_{lr} = \mathbf{x}^T(\mathbf{A}_r)_{:,l}$, $1 \leqslant l \leqslant L_r$, $1 \leqslant r \leqslant R$. Then the quantic can be written as

$$p(\mathbf{y}) = \sum_{r=1}^{R} \sum_{l_1,l_2,l_3=1}^{L_r} (\mathcal{D}_r)_{l_1l_2l_3} \, y_{l_1r} y_{l_2r} y_{l_3r}.$$

In this paper we have presented EVD-based and Kruskal-type conditions guaranteeing essential uniqueness of the decompositions. Important work that remains to be done is the relaxation of the dimensionality constraints on the blocks in the Kruskal-type conditions. Some results based on simultaneous matrix diagonalization are presented in [44]. Also, we have restricted our attention to alternative decompositions that are "nonsingular." We should now check whether, for generic block terms, alternative decompositions in singular terms can exist.

It would be interesting to investigate, given the tensor dimensions I, J, and K, for which block sizes and number of blocks one obtains a generic (in the sense of existing with probability one) or a typical (in the sense of existing with probability different from zero) decomposition. In the context of PARAFAC, generic and typical rank have been studied in [55, 56, 57, 58].

In this paper we limited ourselves to the study of some algebraic aspects of block term decompositions. The computation of the decompositions, by means of alternating least squares algorithms, is addressed in [20]. Some applications are studied in [21, 43, 45].

Acknowledgments. The author wishes to thank A. Stegeman (Heijmans Institute, The Netherlands) and N. Sidiropoulos (Technical University of Crete, Greece)

for sharing the manuscript of [54] before its publication. The author also wishes to thank A. Stegeman for proofreading an early version of the manuscript. A large part of this research was carried out when L. De Lathauwer was with the French Centre National de la Recherche Scientifique (C.N.R.S.).

REFERENCES

[1] C.J. Appellof and E.R. Davidson, *Strategies for analyzing data from video fluoromatric monitoring of liquid chromatographic effluents*, Anal. Chemistry, 53 (1981), pp. 2053–2056.

[2] G. Beylkin and M.J. Mohlenkamp, *Numerical operator calculus in higher dimensions*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 10246–10251.

[3] G. Beylkin and M.J. Mohlenkamp, *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., 26 (2005), pp. 2133–2159.

[4] G. Boutry, M. Elad, G.H. Golub, and P. Milanfar, *The generalized eigenvalue problem for nonsquare pencils using a minimal perturbation approach*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 582–601.

[5] R. Bro, *PARAFAC. Tutorial and Applications*, Chemom. Intell. Lab. Syst., 38 (1997), pp. 149–171.

[6] D. Burdick, X. Tu, L. McGown, and D. Millican, *Resolution of multicomponent fluorescent mixtures by analysis of the excitation-emission-frequency array*, J. Chemometrics, 4 (1990), pp. 15–28.

[7] J. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 9 (1970), pp. 267–283.

[8] P. Comon, *Independent component analysis, a new concept?* Signal Process., 36 (1994), pp. 287–314.

[9] P. Comon and B. Mourrain, *Decomposition of quantics in sums of powers of linear forms*, Signal Process., 53 (1996), pp. 93–108.

[10] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, *Symmetric tensors and symmetric tensor rank*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1254–1279.

[11] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*, Ph.D. thesis, K.U.Leuven, Belgium, 1997.

[12] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[13] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[14] L. De Lathauwer, B. De Moor, and J. Vandewalle, *An introduction to independent component analysis*, J. Chemometrics, 14 (2000), pp. 123–149.

[15] L. De Lathauwer and J. Vandewalle, *Dimensionality reduction in higher-order signal processing and rank-$(R_1, R_2, \ldots, R_N)$ reduction in multilinear algebra*, Linear Algebra Appl., 391 (2004), pp. 31–55.

[16] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Computation of the Canonical Decomposition by means of a simultaneous generalized Schur decompositition*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 295–327.

[17] L. De Lathauwer, *A link between the Canonical Decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666.

[18] L. De Lathauwer and J. Castaing, *Tensor-based techniques for the blind separation of DS-CDMA signals*, Signal Process., 87 (2007), pp. 322–336.

[19] L. De Lathauwer, *Decompositions of a higher-order tensor in block terms—Part* I: *Lemmas for partitioned matrices*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1022–1032.

[20] L. De Lathauwer and D. Nion, *Decompositions of a higher-order tensor in block terms—Part* III: *Alternating Least Squares algorithms*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1067–1083.

[21] L. De Lathauwer and A. de Baynast, *Blind deconvolution of DS-CDMA Signals by means of decomposition in rank-$(1, L, L)$ terms*, IEEE Trans. Signal Process., 56 (2008), pp. 1562–1571.

[22] M. Elad, P. Milanfar, and G.H. Golub, *Shape from moments—an estimation theory perspective*, IEEE Trans. Signal Process., 52 (2004), pp. 1814–1829.

[23] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University

Press, Baltimore, MD, 1996.
[24] W. Hackbush, B.N. Khoromskij, and E. Tyrtyshnikov, *Hierarchical Kronecker tensor-product approximations*, J. Numer. Math., 13 (2005), pp. 119–156.
[25] W. Hackbush and B.N. Khoromskij, *Tensor-product approximation to operators and functions in high dimension*, J. Complexity, 23 (2007), pp. 697–714.
[26] R.A. Harshman, *Foundations of the PARAFAC procedure*: *Model and conditions for an "explanatory" multi-mode factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.
[27] F.L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp. 164–189.
[28] F.L. Hitchcock, *Multiple invariants and generalized rank of a p-way matrix or tensor*, J. Math. Phys., 7 (1927), pp. 39–79.
[29] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley, New York, 2001.
[30] J. Ja'Ja', *An addendum to Kronecker's theory of pencils*, SIAM J. Appl. Math., 37 (1979), pp. 700–712.
[31] T. Jiang and N.D. Sidiropoulos, *Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear models with constant modulus constraints*, IEEE Trans. Signal Process., 52 (2004), pp. 2625–2636.
[32] T. Kato, *A Short Introduction to Perturbation theory for Linear Operators*, Springer-Verlag, New York, 1982.
[33] B.N. Khoromskij and V. Khoromskaia, *Low rank Tucker-type tensor approximation to classical potentials*, Central European J. Math., 5 (2007), pp. 523–550.
[34] H. Kiers, *Towards a standardized notation and terminology in multiway analysis*, J. Chemometrics, 14 (2000), pp. 105–122.
[35] T.G. Kolda, *Orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 243–255.
[36] T.G. Kolda, *A counterexample to the possibility of an extension of the Eckart-Young low-rank approximation theorem for the orthogonal rank tensor decomposition*, SIAM J. Matrix Analysis, 24 (2003), pp. 762–767.
[37] P.M. Kroonenberg, *Applied Multiway Data Analysis*, John Wiley, New York, 2008.
[38] J.B. Kruskal, *Three-way arrays*: *rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.
[39] J.B. Kruskal, *Rank, decomposition, and uniqueness for* 3*-way and N-way arrays*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North–Holland, Amsterdam, 1989, pp. 7–18.
[40] S.E. Leurgans, R.T. Ross, and R.B. Abel, *A decomposition for three-way arrays*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1064–1083.
[41] L.-H. Lim, *Singular values and eigenvalues of tensors: a variational approach*, Proceedings of the First IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2005), Puerto Vallarta, Jalisco State, Mexico, 2005, pp. 129–132.
[42] C.D. Moravitz Martin and C.F. Van Loan, *A Jacobi-type method for computing orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1219–1232.
[43] D. Nion and L. De Lathauwer, *A block factor analysis based receiver for blind multi-user access in wireless communications*, in Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2006), Toulouse, France, 2006, pp. 825–828.
[44] D. Nion and L. De Lathauwer, *A tensor-based blind DS-CDMA receiver using simultaneous matrix diagonalization*, Proceedings of the Eighth IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2007), Helsinki, Finland, 2007.
[45] D. Nion and L. De Lathauwer, *Block component model based blind DS-CDMA receivers*, IEEE Trans. Signal Process., to appear.
[46] C.R. Rao and S.K. Mitra, *Generalized Inverse of Matrices and Its Applications*, John Wiley, New York, 1971.
[47] E. Sanchez and B.R. Kowalski, *Tensorial resolution*: *A direct trilinear decomposition*, J. Chemometrics, 4 (1990), pp. 29–45.
[48] R. Sands and F. Young, *Component models for three-way data*: *An alternating least squares algorithm with optimal scaling features*, Psychometrika, 45 (1980), pp. 39–67.
[49] N. Sidiropoulos, G. Giannakis, and R. Bro, *Blind PARAFAC receivers for DS-CDMA systems*, IEEE Trans. Signal Process., 48 (2000), pp. 810–823.
[50] N. Sidiropoulos, R. Bro, and G. Giannakis, *Parallel factor analysis in sensor array processing*, IEEE Trans. Signal Process., 48 (2000), pp. 2377–2388.
[51] N. Sidiropoulos and R. Bro, *On the uniqueness of multilinear decomposition of N-way*

arrays, J. Chemometrics, 14 (2000), pp. 229–239.
[52] N. Sidiropoulos and G.Z. Dimić, *Blind multiuser detection in W-CDMA systems with large delay spread*, IEEE Signal Process. Letters, 8 (2001), pp. 87–89.
[53] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis. Applications in the Chemical Sciences*, John Wiley, Chichester, UK, 2004.
[54] A. Stegeman and N.D. Sidiropoulos, *On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition*, Linear Algebra Appl., 420 (2007), pp. 540–552.
[55] J.M.F. Ten Berge and H.A.L. Kiers, *Simplicity of core arrays in three-way principal component analysis and the typical rank of $p \times q \times 2$ arrays*, Linear Algebra Appl., 294 (1999), pp. 169–179.
[56] J.M.F. Ten Berge, *The typical rank of tall three-way arrays*, Psychometrika, 65 (2000), pp. 525–532.
[57] J.M.F. Ten Berge, N.D. Sidiropoulos, and R. Rocci, *Typical rank and indscal dimensionality for symmetric three-way arrays of order $I \times 2 \times 2$ or $I \times 3 \times 3$*, Linear Algebra Appl., 388 (2004), pp. 363–377.
[58] J.M.F. Ten Berge, *Simplicity and typical rank of three-way arrays, with applications to Tucker-3 analysis with simple cores*, J. Chem., 18 (2004), pp. 17–21.
[59] L.R. Tucker, *The extension of factor analysis to three-dimensional matrices*, in Contributions to Mathematical Psychology, H. Gulliksen and N. Frederiksen, eds., Holt, Rinehart & Winston, New York, 1964, pp. 109–127.
[60] L.R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
[61] T. Zhang and G.H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1067–1083

DECOMPOSITIONS OF A HIGHER-ORDER TENSOR IN BLOCK TERMS—PART III: ALTERNATING LEAST SQUARES ALGORITHMS*

LIEVEN DE LATHAUWER† AND DIMITRI NION‡

Abstract. In this paper we derive alternating least squares algorithms for the computation of the block term decompositions introduced in Part II. We show that degeneracy can also occur for block term decompositions.

Key words. multilinear algebra, higher-order tensor, Tucker decomposition, canonical decomposition, parallel factors model

AMS subject classifications. 15A18, 15A69

DOI. 10.1137/070690730

1. Introduction.

1.1. Organization of the paper. In the companion paper [11] we introduce decompositions of a higher-order tensor in several types of block terms. In the present paper we propose alternating least squares (ALS) algorithms for the computation of these different decompositions.

In the following subsections we first explain our notation and introduce some basic definitions. In section 1.4 we briefly recall the Tucker decomposition/higher-order singular value decomposition (HOSVD) [40, 41, 6, 7, 8] and also the Canonical/Parallel Factor (CANDECOMP/PARAFAC) decomposition [3, 15] and explain how they can be computed.

In section 2 we present an ALS algorithm for the computation of the decomposition in rank-$(L_r, L_r, 1)$ terms. In section 3 we discuss the decomposition in rank-(L, M, N) terms. Section 4 deals with the type-2 decomposition in rank-$(L, M, \cdot)$ terms. Section 5 is a note on degeneracy.

1.2. Notation. We use $\mathbb{K}$ to denote $\mathbb{R}$ or $\mathbb{C}$ when the difference is not important. In this paper scalars are denoted by lowercase letters (a, b, ...), vectors are written in boldface lowercase ($\mathbf{a}$, $\mathbf{b}$, ...), matrices correspond to boldface capitals ($\mathbf{A}$, $\mathbf{B}$, ...), and tensors are written as calligraphic letters ($\mathcal{A}$, $\mathcal{B}$, ...). This notation is consistently used for lower-order parts of a given structure. For instance, the entry with row index i and column index j in a matrix $\mathbf{A}$, i.e., $(\mathbf{A})_{ij}$, is symbolized by a_{ij} (also $(\mathbf{a})_i = a_i$ and $(\mathcal{A})_{ijk} = a_{ijk}$). If no confusion is possible, the ith column vector of a matrix $\mathbf{A}$

*Received by the editors May 7, 2007; accepted for publication (in revised form) by J. G. Nagy April 14, 2008; published electronically September 25, 2008. This research was supported by Research Council K.U.Leuven: GOA-Ambiorics, CoE EF/05/006 Optimization in Engineering (OPTEC), CIF1; F.W.O.: project G.0321.06 and Research Communities ICCoS, ANMMM, and MLDM; the Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, "Dynamical systems, control and optimization," 2007–2011); and EU: ERNSI.

http://www.siam.org/journals/simax/30-3/69073.html

†Subfaculty Science and Technology, Katholieke Universiteit Leuven Campus Kortrijk, E. Sabbelaan 53, 8500 Kortrijk, Belgium (Lieven.DeLathauwer@kuleuven-kortrijk.be), and Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (Lieven.DeLathauwer@esat.kuleuven.be, http://homes.esat.kuleuven.be/~delathau/home.html).

‡Department of Electronic and Computer Engineering, Technical University of Crete, Kounoupidiana Campus, Chania, Crete, 731 00, Greece (nion@telecom.tuc.gr).

is denoted as $\mathbf{a}_i$, i.e., $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \ldots]$. Sometimes we will use the MATLAB colon notation to indicate submatrices of a given matrix or subtensors of a given tensor. Italic capitals are also used to denote index upper bounds (e.g., $i = 1, 2, \ldots, I$). The symbol $\otimes$ denotes the Kronecker product,

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \ldots \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \ldots \\ \vdots & \vdots & \end{pmatrix}.$$

Let $\mathbf{A} = [\mathbf{A}_1\ \ldots\ \mathbf{A}_R]$ and $\mathbf{B} = [\mathbf{B}_1\ \ldots\ \mathbf{B}_R]$ be two partitioned matrices. Then the Khatri–Rao product is defined as the partitionwise Kronecker product and represented by $\odot$ [34]:

$$\mathbf{A} \odot \mathbf{B} = (\mathbf{A}_1 \otimes \mathbf{B}_1 \ldots \mathbf{A}_R \otimes \mathbf{B}_R). \tag{1.1}$$

In recent years, the term "Khatri–Rao product" and the symbol $\odot$ have mainly been used in the case where $\mathbf{A}$ and $\mathbf{B}$ are partitioned into vectors. For clarity, we denote this particular, columnwise, Khatri–Rao product by $\odot_c$:

$$\mathbf{A} \odot_c \mathbf{B} = (\mathbf{a}_1 \otimes \mathbf{b}_1 \ldots \mathbf{a}_R \otimes \mathbf{b}_R).$$

The superscripts $\cdot^T$, $\cdot^H$, and $\cdot^\dagger$ denote the transpose, complex conjugated transpose, and Moore–Penrose pseudoinverse, respectively. The operator diag$(\cdot)$ stacks its scalar arguments in a square diagonal matrix. Analogously, blockdiag$(\cdot)$ stacks its vector or matrix arguments in a block-diagonal matrix. The $(N \times N)$ identity matrix is represented by $\mathbf{I}_{N\times N}$. $\mathbf{1}_N$ is a column vector of all ones of length N. The zero tensor is denoted by $\mathcal{O}$.

1.3. Basic definitions.

DEFINITION 1.1. *Consider* $\mathcal{T} \in \mathbb{K}^{I_1\times I_2\times I_3}$ *and* $\mathbf{A} \in \mathbb{K}^{J_1\times I_1}$, $\mathbf{B} \in \mathbb{K}^{J_2\times I_2}$, $\mathbf{C} \in \mathbb{K}^{J_3\times I_3}$. *Then the* Tucker mode-1 product $\mathcal{T} \bullet_1 \mathbf{A}$, mode-2 product $\mathcal{T} \bullet_2 \mathbf{B}$, and mode-3 product $\mathcal{T} \bullet_3 \mathbf{C}$ *are defined by*

$$(\mathcal{T} \bullet_1 \mathbf{A})_{j_1 i_2 i_3} = \sum_{i_1=1}^{I_1} t_{i_1 i_2 i_3} a_{j_1 i_1} \qquad \forall j_1, i_2, i_3,$$

$$(\mathcal{T} \bullet_2 \mathbf{B})_{i_1 j_2 i_3} = \sum_{i_2=1}^{I_2} t_{i_1 i_2 i_3} b_{j_2 i_2} \qquad \forall i_1, j_2, i_3,$$

$$(\mathcal{T} \bullet_3 \mathbf{C})_{i_1 i_2 j_3} = \sum_{i_3=1}^{I_3} t_{i_1 i_2 i_3} c_{j_3 i_3} \qquad \forall i_1, i_2, j_3,$$

respectively [5].

In this paper we denote the Tucker mode-n product in the same way as in [4]; in the literature the symbol $\times_n$ is sometimes used [6, 7, 8].

DEFINITION 1.2. *The* Frobenius norm *of a tensor* $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ *is defined as*

$$\|\mathcal{T}\| = \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} |t_{ijk}|^2 \right)^{\frac{1}{2}}.$$

DEFINITION 1.3. *The outer product* $\mathcal{A} \circ \mathcal{B}$ *of a tensor* $\mathcal{A} \in \mathbb{K}^{I_1 \times I_2 \times \ldots \times I_P}$ *and a tensor* $\mathcal{B} \in \mathbb{K}^{J_1 \times J_2 \times \ldots \times J_Q}$ *is the tensor defined by*

$$(\mathcal{A} \circ \mathcal{B})_{i_1 i_2 \ldots i_P j_1 j_2 \ldots j_Q} = a_{i_1 i_2 \ldots i_P} b_{j_1 j_2 \ldots j_Q}$$

for all values of the indices.

For instance, the outer product $\mathcal{T}$ of three vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ is defined by $t_{ijk} = a_i b_j c_k$ for all values of the indices.

DEFINITION 1.4. *A* mode-n vector *of a tensor* $\mathcal{T} \in \mathbb{K}^{I_1 \times I_2 \times I_3}$ *is an* I_n*-dimensional vector obtained from* $\mathcal{T}$ *by varying the index* i_n *and keeping the other indices fixed* [19].

Mode-n vectors generalize column and row vectors.

DEFINITION 1.5. *The* mode-n rank *of a tensor* $\mathcal{A}$ *is the dimension of the subspace spanned by its mode-*n *vectors.*

The mode-n rank of a higher-order tensor is the obvious generalization of the column (row) rank of a matrix.

DEFINITION 1.6. *A third-order tensor is* rank-(L, M, N) *if its mode-*1 *rank, mode-*2 *rank, and mode-*3 *rank are equal to* L, M, *and* N, *respectively.*

A rank-$(1, 1, 1)$ tensor is briefly called rank-1. The rank of a tensor is now defined as follows.

DEFINITION 1.7. *The* rank *of a tensor* $\mathcal{T}$ *is the minimal number of rank-*1 *tensors that yield* $\mathcal{T}$ *in a linear combination* [24].

It will be useful to write tensor expressions in terms of matrices or vectors. We therefore define standard matrix and vector representations of a third-order tensor.

DEFINITION 1.8. *The standard* $(JK \times I)$ *matrix representation* $(\mathcal{T})_{JK \times I} = \mathbf{T}_{JK \times I}$, $(KI \times J)$ *representation* $(\mathcal{T})_{KI \times J} = \mathbf{T}_{KI \times J}$, *and* $(IJ \times K)$ *representation* $(\mathcal{T})_{IJ \times K} = \mathbf{T}_{IJ \times K}$ *of a tensor* $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ *are defined by*

$$\begin{aligned}(\mathbf{T}_{JK \times I})_{(j-1)K+k,i} &= (\mathcal{T})_{ijk},\\ (\mathbf{T}_{KI \times J})_{(k-1)I+i,j} &= (\mathcal{T})_{ijk},\\ (\mathbf{T}_{IJ \times K})_{(i-1)J+j,k} &= (\mathcal{T})_{ijk}\end{aligned}$$

for all values of the indices [19]. *The standard* $(IJK \times 1)$ *vector representation* $(\mathcal{T})_{IJK} = \mathbf{t}_{IJK}$ *of* $\mathcal{T}$ *is defined by*

$$(\mathbf{t}_{IJK})_{(i-1)JK+(j-1)K+k} = (\mathcal{T})_{ijk}$$

for all values of the indices.

Note that in these definitions indices to the right vary more rapidly than indices to the left. Further, the kth $(I \times J)$ matrix slice of $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ will be denoted as $\mathbf{T}_{I \times J,k}$.

1.4. HOSVD and PARAFAC. We have now enough material to introduce the HOSVD [6, 7, 8] and PARAFAC [15] decompositions.

DEFINITION 1.9. *A* HOSVD *of a tensor* $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ *is a decomposition of* $\mathcal{T}$ *of the form*

$$\mathcal{T} = \mathcal{D} \bullet_1 \mathbf{A} \bullet_2 \mathbf{B} \bullet_3 \mathbf{C} \tag{1.2}$$

in which

- *the matrices* $\mathbf{A} \in \mathbb{K}^{I \times L}$, $\mathbf{B} \in \mathbb{K}^{J \times M}$ *and* $\mathbf{C} \in \mathbb{K}^{K \times N}$ *are columnwise orthonormal,*

- *the core tensor* $\mathcal{D} \in \mathbb{K}^{L \times M \times N}$ *is*
  - *all-orthogonal*,

$$\begin{aligned}
\langle \mathbf{D}_{M\times N,l_1}, \mathbf{D}_{M\times N,l_2} \rangle &= \operatorname{trace}(\mathbf{D}_{M\times N,l_1} \cdot \mathbf{D}_{M\times N,l_2}^H) = {\sigma_{l_1}^{(1)}}^2 \delta_{l_1,l_2}, \\
&\qquad 1 \leqslant l_1, l_2 \leqslant L, \\
\langle \mathbf{D}_{N\times L,m_1}, \mathbf{D}_{N\times L,m_2} \rangle &= \operatorname{trace}(\mathbf{D}_{N\times L,m_1} \cdot \mathbf{D}_{N\times L,m_2}^H) = {\sigma_{m_1}^{(2)}}^2 \delta_{m_1,m_2}, \\
&\qquad 1 \leqslant m_1, m_2 \leqslant M, \\
\langle \mathbf{D}_{I\times J,n_1}, \mathbf{D}_{I\times J,n_2} \rangle &= \operatorname{trace}(\mathbf{D}_{L\times M,n_1} \cdot \mathbf{D}_{L\times M,n_2}^H) = {\sigma_{n_1}^{(3)}}^2 \delta_{n_1,n_2}, \\
&\qquad 1 \leqslant n_1, n_2 \leqslant N;
\end{aligned}$$

  - *ordered*,

$$\begin{aligned}
{\sigma_1^{(1)}}^2 &\geqslant {\sigma_2^{(1)}}^2 \geqslant \ldots \geqslant {\sigma_L^{(1)}}^2 \geqslant 0, \\
{\sigma_1^{(2)}}^2 &\geqslant {\sigma_2^{(2)}}^2 \geqslant \ldots \geqslant {\sigma_M^{(2)}}^2 \geqslant 0, \\
{\sigma_1^{(3)}}^2 &\geqslant {\sigma_2^{(3)}}^2 \geqslant \ldots \geqslant {\sigma_N^{(3)}}^2 \geqslant 0.
\end{aligned}$$

Equation (1.2) can be written in terms of the standard $(JK \times I)$, $(KI \times J)$, and $(IJ \times K)$ matrix representations of $\mathcal{T}$ as follows:

$$\mathbf{T}_{JK\times I} = (\mathbf{B} \otimes \mathbf{C}) \cdot \mathbf{D}_{MN\times L} \cdot \mathbf{A}^T, \tag{1.3}$$

$$\mathbf{T}_{KI\times J} = (\mathbf{C} \otimes \mathbf{A}) \cdot \mathbf{D}_{NL\times M} \cdot \mathbf{B}^T, \tag{1.4}$$

$$\mathbf{T}_{IJ\times K} = (\mathbf{A} \otimes \mathbf{B}) \cdot \mathbf{D}_{LM\times N} \cdot \mathbf{C}^T. \tag{1.5}$$

This decomposition is a specific instance of the Tucker decomposition, introduced in [40, 41]; columnwise orthonormality of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and all-orthogonality and ordering of $\mathcal{D}$ were suggested in the computational strategy in [40, 41]. The decomposition exists for any $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$. The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ can be computed as the matrices of right singular vectors associated with the nonzero singular values of $\mathbf{T}_{JK\times I}$, $\mathbf{T}_{KI\times J}$, and $\mathbf{T}_{IJ\times K}$, respectively. The core tensor is then given by $\mathcal{D} = \mathcal{T} \bullet_1 \mathbf{A}^H \bullet_2 \mathbf{B}^H \bullet_3 \mathbf{C}^H$. The values L, M, and N correspond to the rank of $\mathbf{T}_{JK\times I}$, $\mathbf{T}_{KI\times J}$, and $\mathbf{T}_{IJ\times K}$, i.e., they are equal to the mode-1, mode-2, and mode-3 rank of $\mathcal{T}$, respectively. Given the way (1.2) can be computed, it comes as no surprise that the SVD of matrices and the HOSVD of higher-order tensors have some analogous properties [6].

Define $\tilde{\mathcal{D}} = \mathcal{D} \bullet_3 \mathbf{C}$. Then

$$\mathcal{T} = \tilde{\mathcal{D}} \bullet_1 \mathbf{A} \bullet_2 \mathbf{B} \tag{1.6}$$

is a (normalized) *Tucker-2 decomposition* of $\mathcal{T}$.

We are often interested in the best *approximation* of a given tensor $\mathcal{T}$ by a tensor of which the mode-1 rank, mode-2 rank, and mode-3 rank are upper-bounded by L, M, and N, respectively. Formally, we want to find $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathcal{D})$ such that $\hat{\mathcal{T}} = \mathcal{D} \bullet_1 \mathbf{A} \bullet_2 \mathbf{B} \bullet_3 \mathbf{C}$ minimizes the least-squares cost function $f(\hat{\mathcal{T}}) = \|\mathcal{T} - \hat{\mathcal{T}}\|^2$. One difference between matrices and tensors is that this optimal approximation cannot in general be obtained by simple truncation of the HOSVD. The algorithms discussed in [7, 8, 14, 17, 20, 21, 22, 23, 44] aim at finding the optimal approximation. These algorithms can be initialized with the approximation obtained by truncation.

Besides the HOSVD, there exist other ways to generalize the SVD of matrices. The most well known is CANDECOMP/PARAFAC [3, 15].

DEFINITION 1.10. *A* canonical or parallel factor decomposition (CANDECOMP/PARAFAC) *of a tensor* $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ *is a decomposition of* $\mathcal{T}$ *as a linear combination of rank-*1 *terms*:

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{1.7}$$

In terms of the standard matrix representations of $\mathcal{T}$, decomposition (1.7) can be written as

$$\mathbf{T}_{JK\times I} = (\mathbf{B} \odot_c \mathbf{C}) \cdot \mathbf{A}^T, \tag{1.8}$$

$$\mathbf{T}_{KI\times J} = (\mathbf{C} \odot_c \mathbf{A}) \cdot \mathbf{B}^T, \tag{1.9}$$

$$\mathbf{T}_{IJ\times K} = (\mathbf{A} \odot_c \mathbf{B}) \cdot \mathbf{C}^T. \tag{1.10}$$

In terms of the $(IJK\times 1)$ vector representation of $\mathcal{T}$, the decomposition can be written as

$$\mathbf{T}_{IJK} = (\mathbf{A} \odot_c \mathbf{B} \odot_c \mathbf{C}) \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{1.11}$$

PARAFAC components are usually estimated by minimization of the quadratic cost function

$$f(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \|\mathcal{T} - \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r\|^2. \tag{1.12}$$

This is most often done by means of an ALS algorithm, in which the vectors are updated mode per mode [3, 37]. Since PARAFAC is trilinear in its arguments, updating $\mathbf{A}$, given $\mathbf{B}$ and $\mathbf{C}$, is just a linear least squares problem. The same holds for updating $\mathbf{B}$, given $\mathbf{A}$ and $\mathbf{C}$, and updating $\mathbf{C}$, given $\mathbf{A}$ and $\mathbf{B}$. The algorithm is outlined in Table 1.1. The normalization of $\mathbf{B}$ and $\mathbf{C}$, in steps 2 and 3, respectively, is meant to avoid over- and underflow. Scaling factors are absorbed in the matrix $\mathbf{A}$. Note that the matrices $\mathbf{B} \odot_c \mathbf{C}$, $\mathbf{C} \odot_c \mathbf{A}$, and $\mathbf{A} \odot_c \mathbf{B}$ have to have at least as many rows as columns and that they have to be full column rank.

ALS iterations are sometimes slow. In addition, it is sometimes observed that the algorithm moves through a "swamp": the algorithm seems to converge, but then the convergence speed drastically decreases and remains small for several iteration steps, after which it may suddenly increase again. Recently, it has been understood that the multilinearity of PARAFAC allows for the determination of the optimal step size, which improves convergence [33].

In many applications one can assume that $\mathbf{A}$ and $\mathbf{B}$ are full column rank (this implies that $R \leqslant \min(I, J)$) and that $\mathbf{C}$ does not contain collinear vectors. Assume for convenience that the values $c_{21}, \ldots, c_{2R}$ are nonzero, such that $\mathbf{T}_{I\times J,2}$ is rank-R, and that the values $c_{11}/c_{21}, \ldots, c_{1R}/c_{2R}$ are mutually different. (If this is not the case, then we can consider linear combinations of slices such that the following reasoning applies.) Then $\mathbf{A}$ follows from the eigenvalue decomposition (EVD) $\mathbf{T}_{I\times J,1} \cdot \mathbf{T}^{\dagger}_{I\times J,2} = \mathbf{A} \cdot \mathrm{diag}(c_{11}/c_{21}, \ldots, c_{1R}/c_{2R}) \cdot \mathbf{A}^{\dagger}$. In other words, the columns of $(\mathbf{A}^T)^{\dagger}$ are generalized eigenvectors of the pencil $(\mathbf{T}^T_{I\times J,1}, \mathbf{T}^T_{I\times J,2})$; see [1, 13] and references therein. After having found $\mathbf{A}$, matrix $\mathbf{B}$ may, up to a scaling of its columns, be

TABLE 1.1
ALS algorithm for CANDECOMP/PARAFAC.

- Initialize $\mathbf{B}$, $\mathbf{C}$
- Iterate until convergence:
  1. Update $\mathbf{A}$:
  $$\mathbf{A} \leftarrow \left[(\mathbf{B} \odot_c \mathbf{C})^\dagger \cdot \mathbf{T}_{JK\times I}\right]^T$$
  2. Update $\mathbf{B}$:
  $$\tilde{\mathbf{B}} = \left[(\mathbf{C} \odot_c \mathbf{A})^\dagger \cdot \mathbf{T}_{KI\times J}\right]^T$$
  For $r = 1, \ldots, R$: $\mathbf{b}_r \leftarrow \tilde{\mathbf{b}}_r / \|\tilde{\mathbf{b}}_r\|$
  3. Update $\mathbf{C}$:
  $$\tilde{\mathbf{C}} = \left[(\mathbf{A} \odot_c \mathbf{B})^\dagger \cdot \mathbf{T}_{IJ\times K}\right]^T$$
  For $r = 1, \ldots, R$: $\mathbf{c}_r \leftarrow \tilde{\mathbf{c}}_r / \|\tilde{\mathbf{c}}_r\|$

obtained from $(\mathbf{A}^\dagger \cdot \mathbf{T}_{I\times J,2})^T = \mathbf{B} \cdot \mathrm{diag}(c_{21}, \ldots, c_{2R})$. Matrix $\mathbf{C}$ may then be computed as $\left[(\mathbf{A} \odot_c \mathbf{B})^\dagger \cdot \mathbf{T}_{IJ\times K}\right]^T$. The EVD solution may subsequently be used to initialize the ALS algorithm. This approach has been proposed in [2, 26, 35, 36].

From a numerical point of view, it is preferable to take all the matrix slices of $\mathcal{T}$ into account, instead of only two of them. We therefore proposed to compute the solution by means of simultaneous matrix diagonalization in [9]. It was shown in [10] that the solution can still be obtained by means of a simultaneous matrix diagonalization when $\mathcal{T}$ is tall in its third mode (meaning that $R \leqslant K$) and $R(R-1) \leqslant I(I-1)J(J-1)/2$.

In [32] a Gauss–Newton method is described, in which all the factors are updated simultaneously; in addition, the inherent indeterminacy of the decomposition has been fixed by adding a quadratic regularization constraint on the component entries. Instead of the least squares error (1.12), one can also minimize the least absolute error. To this end, an alternating linear programming algorithm as well as a weighted median filtering iteration are derived in [42].

2. Decomposition in rank-$(L_r, L_r, 1)$ terms.

2.1. Definition.

DEFINITION 2.1. *A* decomposition of a tensor $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ in a sum of rank-$(L_r, L_r, 1)$ terms, $1 \leqslant r \leqslant R$, *is a decomposition of* $\mathcal{T}$ *of the form*

$$\mathcal{T} = \sum_{r=1}^{R} (\mathbf{A}_r \cdot \mathbf{B}_r^T) \circ \mathbf{c}_r, \tag{2.1}$$

in which the matrix $\mathbf{A}_r \in \mathbb{K}^{I\times L_r}$ *and the matrix* $\mathbf{B}_r \in \mathbb{K}^{J\times L_r}$ *are rank-*L_r, $1 \leqslant r \leqslant R$.

Define $\mathbf{A} = [\mathbf{A}_1 \ldots \mathbf{A}_R]$, $\mathbf{B} = [\mathbf{B}_1 \ldots \mathbf{B}_R]$, $\mathbf{C} = [\mathbf{c}_1 \ldots \mathbf{c}_R]$. In terms of the standard matrix representations of $\mathcal{T}$, (2.1) can be written as

$$\mathbf{T}_{IJ\times K} = [(\mathbf{A}_1 \odot_c \mathbf{B}_1)\mathbf{1}_{L_1} \ \ldots \ (\mathbf{A}_R \odot_c \mathbf{B}_R)\mathbf{1}_{L_R}] \cdot \mathbf{C}^T, \tag{2.2}$$

$$\mathbf{T}_{JK\times I} = (\mathbf{B} \odot \mathbf{C}) \cdot \mathbf{A}^T, \tag{2.3}$$

$$\mathbf{T}_{KI\times J} = (\mathbf{C} \odot \mathbf{A}) \cdot \mathbf{B}^T. \tag{2.4}$$

TABLE 2.1
ALS algorithm for decomposition in rank-$(L_r, L_r, 1)$ terms.

- Initialize $\mathbf{B}$, $\mathbf{C}$
- Iterate until convergence:
  1. Update $\mathbf{A}$:
$$\mathbf{A} \leftarrow \left[(\mathbf{B} \odot \mathbf{C})^\dagger \cdot \mathbf{T}_{JK \times I}\right]^T$$
  2. Update $\mathbf{B}$:
$$\tilde{\mathbf{B}} = \left[(\mathbf{C} \odot \mathbf{A})^\dagger \cdot \mathbf{T}_{KI \times J}\right]^T$$
  For $r = 1, \ldots, R$: QR-factorization: $\tilde{\mathbf{B}}_r = \mathbf{QR}$, $\mathbf{B}_r \leftarrow \mathbf{Q}$
  3. Update $\mathbf{C}$:
$$\tilde{\mathbf{C}} = \left\{[(\mathbf{A}_1 \odot_c \mathbf{B}_1)\mathbf{1}_{L_1} \ \ldots \ (\mathbf{A}_R \odot_c \mathbf{B}_R)\mathbf{1}_{L_R}]^\dagger \cdot \mathbf{T}_{IJ \times K}\right\}^T$$
  For $r = 1, \ldots, R$: $\mathbf{c}_r \leftarrow \tilde{\mathbf{c}}_r / \|\tilde{\mathbf{c}}_r\|$

2.2. Algorithm. Like PARAFAC, the decomposition in rank-$(L_r, L_r, 1)$ terms is trilinear in the component matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. This means that updating $\mathbf{A}$, given $\mathbf{B}$ and $\mathbf{C}$, is just a linear least squares problem. The same holds for updating $\mathbf{B}$, given $\mathbf{A}$ and $\mathbf{C}$, and updating $\mathbf{C}$, given $\mathbf{A}$ and $\mathbf{B}$. The update rules follow directly from (2.2)–(2.4). The algorithm is outlined in Table 2.1. The normalization in steps 2 and 3 are meant to avoid under- and overflow. Moreover, the normalization in step 2 prevents the submatrices of $\mathbf{B}$ from becoming ill-conditioned. Analogous to the situation for PARAFAC, the matrices $\mathbf{B} \odot_c \mathbf{C}$, $\mathbf{C} \odot_c \mathbf{A}$, and $[(\mathbf{A}_1 \odot_c \mathbf{B}_1)\mathbf{1}_{L_1} \ \ldots \ (\mathbf{A}_R \odot_c \mathbf{B}_R)\mathbf{1}_{L_R}]$ have to have at least as many rows as columns and have to be full column rank.

If $\mathbf{A}$ and $\mathbf{B}$ are full column rank and $\mathbf{C}$ does not have collinear vectors, then this algorithm may be initialized by means of a (generalized) EVD, as explained in the proof of [11, Theorem 4.1].

2.3. Numerical experiments. We generate tensors $\tilde{\mathcal{T}} \in \mathbb{C}^{5 \times 6 \times 5}$ in the following way:

$$\tilde{\mathcal{T}} = \frac{\mathcal{T}}{\|\mathcal{T}\|} + \sigma_N \frac{\mathcal{N}}{\|\mathcal{N}\|}, \tag{2.5}$$

in which $\mathcal{T}$ can be decomposed as in (2.1). We consider $R = 3$ rank-$(2, 2, 1)$ terms, i.e., $\mathbf{A}_r \in \mathbb{C}^{5 \times 2}$, $\mathbf{B}_r \in \mathbb{C}^{6 \times 2}$, $\mathbf{C}_r \in \mathbb{C}^{6 \times 1}$, $1 \leqslant r \leqslant 3$. The decomposition of $\mathcal{T}$ is essentially unique by [11, Theorem 4.4]. The second term in (2.5) is a noise term. The entries of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathcal{N}$ are drawn from a zero-mean unit-variance Gaussian distribution. The parameter σ_N controls the noise level.

A Monte Carlo experiment consisting of 200 runs was carried out. The algorithm was initialized with three random starting values.

The accuracy is measured in terms of the relative error $e = \|\mathbf{C} - \hat{\mathbf{C}}\|/\|\mathbf{C}\|$, in which $\hat{\mathbf{C}}$ is the estimate of $\mathbf{C}$, optimally ordered and scaled. The median results are plotted in Figure 2.1. We plot the median instead of the mean because, in some of the

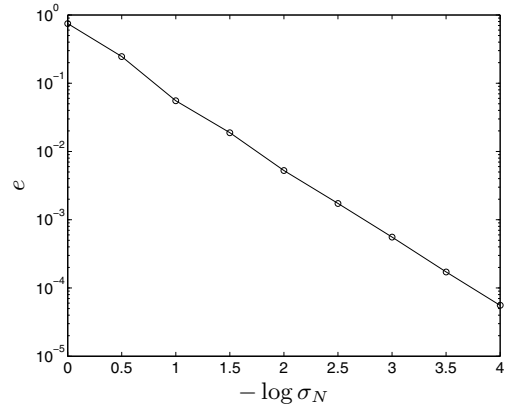


FIG. 2.1. *Median relative error obtained in the first experiment in section* 2.3.

runs, the convergence became too slow for the algorithm to find a sufficiently accurate estimate in a reasonable time.

In a second experiment, we generate tensors $\tilde{\mathcal{T}} \in \mathbb{C}^{10\times 10\times 10}$ as in (2.5). We consider $R = 5$ rank-$(2,2,1)$ terms, i.e., $\mathbf{A}_r \in \mathbb{C}^{10\times 2}$, $\mathbf{B}_r \in \mathbb{C}^{10\times 2}$, $\mathbf{C}_r \in \mathbb{C}^{10\times 1}$, $1 \leqslant r \leqslant 5$. The five rank-$(2,2,1)$ terms are scaled such that their Frobenius norm equals 1, 3.25, 5.5, 7.75, and 10, respectively. The fact that there is a difference of 20 dB between the strongest and the weakest term makes this problem quite hard. The decomposition of $\mathcal{T}$ is essentially unique by [11, Theorem 4.1]. In Figure 2.2 we show the median accuracy obtained when the algorithm in Table 2.1 is initialized (i) by means of a (generalized) EVD, as explained in the proof of [11, Theorem 4.1], and (ii) by means of a random starting value. It is clear that the global optimum is not found when the algorithm is initialized randomly. However, the initialization by means of a (generalized) EVD does lead to the global solution when the signal-to-noise ratio (SNR) is sufficiently high. As a matter of fact, the (generalized) EVD yields the exact solution when the data are noise-free.

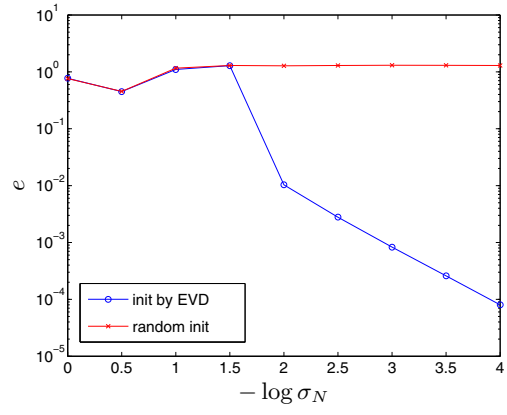


FIG. 2.2. *Median relative error obtained in the second experiment in section* 2.3.

3. Decomposition in rank-(L, M, N) terms.

3.1. Definition.

DEFINITION 3.1. *A* decomposition of a tensor $\mathcal{T} \in \mathbb{K}^{I\times J\times K}$ in a sum of rank-(L, M, N) terms *is a decomposition of* $\mathcal{T}$ *of the form*

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r \bullet_3 \mathbf{C}_r, \tag{3.1}$$

in which $\mathcal{D}_r \in \mathbb{K}^{L\times M\times N}$ *are full rank-*(L, M, N) *and in which* $\mathbf{A}_r \in \mathbb{K}^{I\times L}$ *(with* $I \geqslant L$*),* $\mathbf{B}_r \in \mathbb{K}^{J\times M}$ *(with* $J \geqslant M$*), and* $\mathbf{C}_r \in \mathbb{K}^{K\times N}$ *(with* $K \geqslant N$*) are full column rank,* $1 \leqslant r \leqslant R$*.*

Define partitioned matrices $\mathbf{A} = [\mathbf{A}_1 \dots \mathbf{A}_R]$, $\mathbf{B} = [\mathbf{B}_1 \dots \mathbf{B}_R]$, and $\mathbf{C} = [\mathbf{C}_1 \dots \mathbf{C}_R]$. In terms of the standard matrix representations of $\mathcal{T}$, (3.1) can be written as

$$\mathbf{T}_{JK\times I} = (\mathbf{B} \odot \mathbf{C}) \cdot \text{blockdiag}((\mathcal{D}_1)_{MN\times L}, \dots, (\mathcal{D}_R)_{MN\times L}) \cdot \mathbf{A}^T, \tag{3.2}$$

$$\mathbf{T}_{KI\times J} = (\mathbf{C} \odot \mathbf{A}) \cdot \text{blockdiag}((\mathcal{D}_1)_{NL\times M}, \dots, (\mathcal{D}_R)_{NL\times M}) \cdot \mathbf{B}^T, \tag{3.3}$$

$$\mathbf{T}_{IJ\times K} = (\mathbf{A} \odot \mathbf{B}) \cdot \text{blockdiag}((\mathcal{D}_1)_{LM\times N}, \dots, (\mathcal{D}_R)_{LM\times N}) \cdot \mathbf{C}^T. \tag{3.4}$$

In terms of the $(IJK\times 1)$ vector representation of $\mathcal{T}$, the decomposition can be written as

$$\mathbf{t}_{IJK} = (\mathbf{A} \odot \mathbf{B} \odot \mathbf{C}) \cdot \begin{pmatrix} (\mathcal{D}_1)_{LMN} \\ \vdots \\ (\mathcal{D}_R)_{LMN} \end{pmatrix}. \tag{3.5}$$

3.2. Algorithm. The decomposition in rank-(L, M, N) terms is quadrilinear in its factors $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathcal{D}$. Hence, the conditional update of $\mathbf{A}$, given $\mathbf{B}$, $\mathbf{C}$, and $\mathcal{D}$, is a linear least squares problem. The same holds for conditional updates of $\mathbf{B}$, $\mathbf{C}$, and $\mathcal{D}$. The update rules follow directly from (3.2)–(3.5). The algorithm is outlined in Table 3.1. This algorithm is a generalization of the algorithm in [43] for the computation of the best rank-(L, M, N) approximation of a given tensor. The matrices $(\mathbf{B} \odot \mathbf{C}) \cdot \text{blockdiag}((\mathcal{D}_1)_{MN\times L}, \dots, (\mathcal{D}_R)_{MN\times L})$, $(\mathbf{C} \odot \mathbf{A}) \cdot \text{blockdiag}((\mathcal{D}_1)_{NL\times M}, \dots, (\mathcal{D}_R)_{NL\times M})$, and $(\mathbf{A} \odot \mathbf{B}) \cdot \text{blockdiag}((\mathcal{D}_1)_{LM\times N}, \dots, (\mathcal{D}_R)_{LM\times N})$ have to have at least as many rows as columns and have to be full column rank.

The order of the updates in Table 3.1 is not mandatory. We have observed in numerical experiments that it is often advantageous to alternate between a few updates of $\mathbf{A}$ and $\mathcal{D}$, then alternate between a few updates of $\mathbf{B}$ and $\mathcal{D}$, and so on.

3.3. Numerical experiments. We generate tensors $\tilde{\mathcal{T}} \in \mathbb{C}^{5\times 5\times 7}$ as in (2.5). The tensors $\mathcal{T}$ can now be decomposed as in (3.1). We consider $R = 2$ terms characterized by $\mathbf{A}_r \in \mathbb{C}^{5\times 2}$, $\mathbf{B}_r \in \mathbb{C}^{5\times 2}$, $\mathbf{C}_r \in \mathbb{C}^{7\times 3}$, and $\mathcal{D}_r \in \mathbb{C}^{2\times 2\times 3}$, $1 \leqslant r \leqslant 2$. The entries of $\mathbf{A}_r$, $\mathbf{B}_r$, $\mathbf{C}_r$, $\mathcal{D}_r$, and $\mathcal{N}$ are drawn from a zero-mean unit-variance Gaussian distribution. The decomposition of $\mathcal{T}$ is essentially unique by [11, Theorem 5.1].

A Monte Carlo experiment consisting of 200 runs was carried out. The algorithm was initialized with three random starting values.

The accuracy is measured in terms of the relative error $e = \|\mathbf{C}-\hat{\mathbf{C}}\|/\|\mathbf{C}\|$, in which $\hat{\mathbf{C}}$ is the estimate of $\mathbf{C}$, of which the submatrices are optimally ordered and multiplied from the right by a (3×3) matrix. The median results are plotted in Figure 3.1.

Next, we check what happens if the algorithm in Table 3.1 is used for the computation of the decomposition in rank-$(L, L, 1)$ terms. In this case, the tensors $\mathcal{D}_r$ are of

TABLE 3.1
ALS algorithm for decomposition in rank-(L, M, N) terms.

- Initialize $\mathbf{B}$, $\mathbf{C}$, $\mathcal{D}$
- Iterate until convergence:
  1. Update $\mathbf{A}$:

$$\tilde{\mathbf{A}} = \left[\text{blockdiag}((\mathcal{D}_1)^{\dagger}_{MN\times L}, \ldots, (\mathcal{D}_R)^{\dagger}_{MN\times L}) \cdot (\mathbf{B} \odot \mathbf{C})^{\dagger} \cdot \mathbf{T}_{JK\times I}\right]^T$$

  For $r = 1, \ldots, R$: QR-factorization: $\tilde{\mathbf{A}}_r = \mathbf{QR}$, $\mathbf{A}_r \leftarrow \mathbf{Q}$
  2. Update $\mathbf{B}$:

$$\tilde{\mathbf{B}} = \left[\text{blockdiag}((\mathcal{D}_1)^{\dagger}_{NL\times M}, \ldots, (\mathcal{D}_R)^{\dagger}_{NL\times M}) \cdot (\mathbf{C} \odot \mathbf{A})^{\dagger} \cdot \mathbf{T}_{KI\times J}\right]^T$$

  For $r = 1, \ldots, R$: QR-factorization: $\tilde{\mathbf{B}}_r = \mathbf{QR}$, $\mathbf{B}_r \leftarrow \mathbf{Q}$
  3. Update $\mathbf{C}$:

$$\tilde{\mathbf{C}} = \left[\text{blockdiag}((\mathcal{D}_1)^{\dagger}_{LM\times N}, \ldots, (\mathcal{D}_R)^{\dagger}_{LM\times N}) \cdot (\mathbf{A} \odot \mathbf{B})^{\dagger} \cdot \mathbf{T}_{IJ\times K}\right]^T$$

  For $r = 1, \ldots, R$: QR-factorization: $\tilde{\mathbf{C}}_r = \mathbf{QR}$, $\mathbf{C}_r \leftarrow \mathbf{Q}$
  4. Update $\mathcal{D}$:

$$\begin{pmatrix} (\mathcal{D}_1)_{LMN} \\ \vdots \\ (\mathcal{D}_R)_{LMN} \end{pmatrix} \leftarrow (\mathbf{A} \odot \mathbf{B} \odot \mathbf{C})^{\dagger} \cdot \mathbf{t}_{IJK}$$

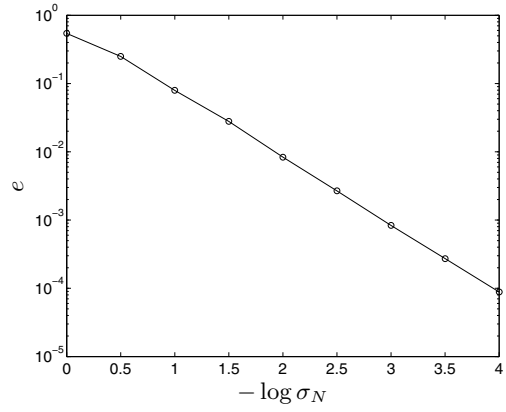


FIG. 3.1. *Median relative error obtained in the first experiment in section* 3.3.

dimension $(L \times L \times 1)$. The data are generated as in the first experiment in section 2.3. We compare three algorithms: (i) the algorithm of Table 2.1, which we denote as Alg $(L, L, 1)$, (ii) the algorithm of Table 3.1, which we denote as Alg (L, M, N), and (iii) a variant of the algorithm of Table 3.1 in which one alternates between a few updates of $\mathbf{A}$ and $\mathbf{D}$, then alternates between a few updates of $\mathbf{B}$ and $\mathbf{D}$, and so on, as explained at the end of section 3.2. The latter algorithm is denoted as Alg $(L, M, N)^*$. The inner iteration is terminated if the Frobenius norm of the difference between two consecutive approximations of $\mathcal{T}$ drops below $1e-6$, with a maximum of 10 inner iterations. We observed that most of the time not more than two or three inner iterations were carried out. We computed the results for one and two random initializations, respectively.

The median results for accuracy and computation time are plotted in Figures 3.2 and 3.3, respectively. From Figure 3.2 it is clear that Alg (L, M, N) does not find the global optimum if it is initialized only once. One should perform inner iterations, or initialize several times. However, both remedies increase the computational cost, as is clear from Figure 3.3. Given that Alg (L, M, N) is by itself more expensive than Alg $(L, L, 1)$, we conclude that it is advantageous to compute the decomposition in rank-$(L, L, 1)$ terms by means of Alg $(L, L, 1)$.

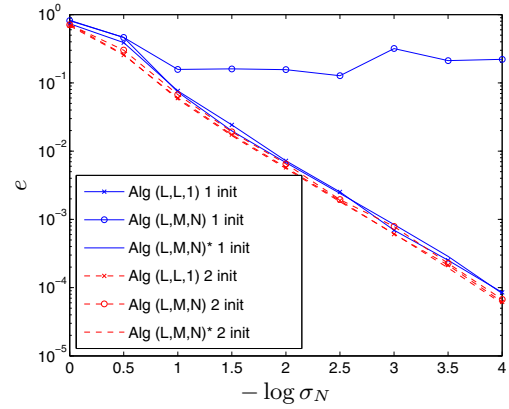


FIG. 3.2. *Median relative error obtained in the second experiment in section* 3.3.

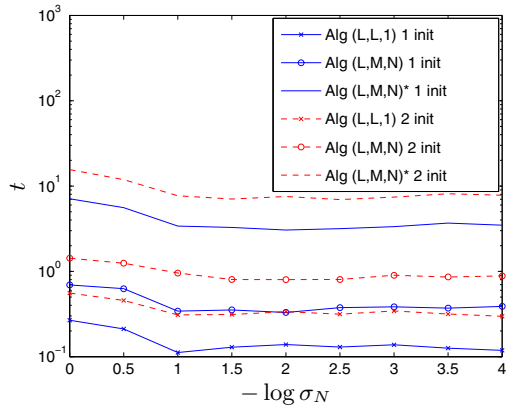


FIG. 3.3. *Median computation time in the second experiment in section* 3.3.

4. Type-2 decomposition in rank-$(L, M, \cdot)$ terms.

4.1. Definition.

DEFINITION 4.1. *A* type-2 decomposition of a tensor $\mathcal{T} \in \mathbb{K}^{I \times J \times K}$ in a sum of rank-$(L, M, \cdot)$ terms *is a decomposition of* $\mathcal{T}$ *of the form*

$$\mathcal{T} = \sum_{r=1}^{R} \mathcal{C}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r, \tag{4.1}$$

in which $\mathcal{C}_r \in \mathbb{K}^{L \times M \times K}$ *(with mode-1 rank equal to* L *and mode-2 rank equal to* M*), and in which* $\mathbf{A}_r \in \mathbb{K}^{I \times L}$ *(with* $I \geqslant L$*) and* $\mathbf{B}_r \in \mathbb{K}^{J \times M}$ *(with* $J \geqslant M$*) are full column rank,* $1 \leqslant r \leqslant R$.

TABLE 4.1
ALS algorithm for type-2 decomposition in rank-$(L, M, \cdot)$ terms.

- Initialize $\mathbf{B}, \mathcal{C}_1, \ldots, \mathcal{C}_R$
- Iterate until convergence:
  1. Update $\mathbf{A}$:

$$\tilde{\mathbf{A}} = \left\{ [(\mathcal{C}_1 \bullet_2 \mathbf{B}_1)_{JK\times L} \ \ldots \ (\mathcal{C}_R \bullet_2 \mathbf{B}_R)_{JK\times L}]^\dagger \cdot \mathbf{T}_{JK\times I} \right\}^T$$

  For $r = 1, \ldots, R$: QR-factorization: $\tilde{\mathbf{A}}_r = \mathbf{QR}$, $\mathbf{A}_r \leftarrow \mathbf{Q}$
  2. Update $\mathbf{B}$:

$$\tilde{\mathbf{B}} = \left\{ [(\mathcal{C}_1 \bullet_1 \mathbf{A}_1)_{KI\times M} \ \ldots \ (\mathcal{C}_R \bullet_1 \mathbf{A}_R)_{KI\times M})]^\dagger \cdot \mathbf{T}_{KI\times J} \right\}^T$$

  For $r = 1, \ldots, R$: QR-factorization: $\tilde{\mathbf{B}}_r = \mathbf{QR}$, $\mathbf{B}_r \leftarrow \mathbf{Q}$
  3. Update $\mathcal{C}_1, \ldots, \mathcal{C}_R$:

$$\begin{pmatrix} (\mathcal{C}_1)_{(LM\times K)} \\ \vdots \\ (\mathcal{C}_R)_{(LM\times K)} \end{pmatrix} \leftarrow (\mathbf{A} \odot \mathbf{B})^\dagger \cdot \mathbf{T}_{IJ\times K}$$

Define partitioned matrices $\mathbf{A} = [\mathbf{A}_1 \ldots \mathbf{A}_R]$ and $\mathbf{B} = [\mathbf{B}_1 \ldots \mathbf{B}_R]$. In terms of the standard matrix representations of $\mathcal{T}$, (4.1) can be written as

$$\mathbf{T}_{IJ\times K} = (\mathbf{A} \odot \mathbf{B}) \cdot \begin{pmatrix} (\mathcal{C}_1)_{(LM\times K)} \\ \vdots \\ (\mathcal{C}_R)_{(LM\times K)} \end{pmatrix}, \tag{4.2}$$

$$\mathbf{T}_{JK\times I} = [(\mathcal{C}_1 \bullet_2 \mathbf{B}_1)_{JK\times L} \ \ldots \ (\mathcal{C}_R \bullet_2 \mathbf{B}_R)_{JK\times L}] \cdot \mathbf{A}^T, \tag{4.3}$$

$$\mathbf{T}_{KI\times J} = [(\mathcal{C}_1 \bullet_1 \mathbf{A}_1)_{KI\times M} \ \ldots \ (\mathcal{C}_R \bullet_1 \mathbf{A}_R)_{KI\times M}] \cdot \mathbf{B}^T. \tag{4.4}$$

4.2. Algorithm. Since the type-2 decomposition in rank-$(L, M, \cdot)$ terms is trilinear in $\mathbf{A}$, $\mathbf{B}$, and $\mathcal{C}$, an ALS algorithm consists of successive linear least squares problems. The update rules for $\mathbf{A}$, $\mathbf{B}$, and $\mathcal{C}$ follow directly from (4.3), (4.4), and (4.2), respectively. The algorithm is outlined in Table 4.1. The matrices $\mathbf{A} \odot \mathbf{B}$, $[(\mathcal{C}_1 \bullet_2 \mathbf{B}_1)_{JK\times L} \ \ldots \ (\mathcal{C}_R \bullet_2 \mathbf{B}_R)_{JK\times L}]$, and $[(\mathcal{C}_1 \bullet_1 \mathbf{A}_1)_{KI\times M} \ \ldots \ (\mathcal{C}_R \bullet_1 \mathbf{A}_R)_{KI\times M}]$ have to have at least as many rows as columns and have to be full column rank.

4.3. Numerical experiment. We generate tensors $\tilde{\mathcal{T}} \in \mathbb{C}^{5\times 6\times 6}$ as in (2.5). The tensors $\mathcal{T}$ can now be decomposed as in (4.1). We consider $R = 3$ terms characterized by $\mathbf{A}_r \in \mathbb{C}^{5\times 2}$, $\mathbf{B}_r \in \mathbb{C}^{6\times 2}$, and $\mathcal{C}_r \in \mathbb{C}^{2\times 2\times 6}$, $1 \leqslant r \leqslant 3$. The entries of $\mathbf{A}_r$, $\mathbf{B}_r$, $\mathbf{C}_r$, and $\mathcal{N}$ are drawn from a zero-mean unit-variance Gaussian distribution. The decomposition of $\mathcal{T}$ is essentially unique by [11, Example 3].

A Monte Carlo experiment consisting of 200 runs was carried out. The algorithm was initialized with three random starting values.

The accuracy is measured in terms of the relative error $e = \|\mathbf{B} - \hat{\mathbf{B}}\|/\|\mathbf{B}\|$, in which $\hat{\mathbf{B}}$ is the estimate of $\mathbf{B}$, of which the submatrices are optimally ordered and multiplied from the right by a (2×2) matrix. The median results are plotted in Figure 4.1.

5. Degeneracy. In the real field, PARAFAC algorithms sometimes show the following behavior. The norm of individual terms in (1.12) goes to infinity, but these terms almost completely cancel each other, such that the overall error continues to

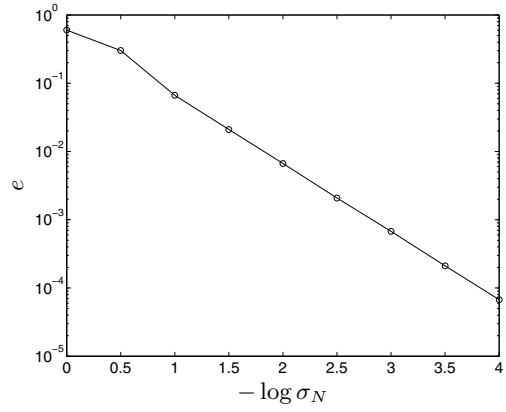


FIG. 4.1. *Median relative error obtained in the experiment in section* 4.3.

decrease. This phenomenon is known as "degeneracy" [16, 25, 27]. It is caused by the fact that for $\tilde{R} \geqslant 2$, the set

$$\mathsf{U}_{\tilde{R}} = \{\mathcal{T} \in \mathbb{R}^{I \times J \times K} | \text{rank}(\mathcal{T}) \leqslant \tilde{R}\}$$

is not closed [12, 25, 38]. The set of tensors that are the sum of at most $\tilde{R} \geqslant 2$ rank-(L, M, N) terms,

$$\mathsf{V}_{\tilde{R}} = \{\mathcal{T} \in \mathbb{R}^{I \times J \times K} | \mathcal{T} \text{ decomposable as in (3.1), with } R \leqslant \tilde{R} \text{ and } \tilde{R} \geqslant 2\},$$

is not closed either. We give an explicit example that is a straightforward generalization of the example given for PARAFAC in [12]. Analogous results hold for the other types of block term decompositions.

Let $\mathbf{I}_1 \in \mathbb{R}^{4 \times 2}$ and $\mathbf{I}_2 \in \mathbb{R}^{4 \times 2}$ consist of the first (resp., last) two columns of $\mathbf{I}_{4 \times 4}$. Consider the tensor $\mathcal{E} \in \mathbb{R}^{2 \times 2 \times 2}$ defined by

$$\begin{aligned} &e_{111} = e_{221} = e_{122} = 1, \\ &e_{121} = e_{211} = e_{112} = e_{212} = e_{222} = 0. \end{aligned}$$

This tensor is rank-3 in $\mathbb{R}$; see [5, pp. 21–22] and [18, section 3]. Now define $\mathcal{T} \in \mathbb{R}^{4 \times 4 \times 4}$ as follows:

$$\begin{aligned} \mathcal{T}(1:2, 1:2, 1:2) &= \mathcal{T}(3:4, 3:4, 1:2) = \mathcal{T}(1:2, 3:4, 3:4) = \mathcal{E}, \\ \mathcal{T}(3:4, 1:2, 1:2) &= \mathcal{T}(3:4, 1:2, 3:4) = \mathcal{T}(1:2, 3:4, 1:2) \\ &= \mathcal{T}(1:2, 1:2, 3:4) = \mathcal{T}(3:4, 3:4, 3:4) = \mathcal{O}_{2 \times 2 \times 2}. \end{aligned}$$

This tensor can be decomposed in three rank-$(2, 2, 2)$ terms:

$$\mathcal{T} = \mathcal{E} \bullet_1 \mathbf{I}_1 \bullet_2 \mathbf{I}_1 \bullet_3 \mathbf{I}_1 + \mathcal{E} \bullet_1 \mathbf{I}_1 \bullet_2 \mathbf{I}_2 \bullet_3 \mathbf{I}_2 + \mathcal{E} \bullet_1 \mathbf{I}_2 \bullet_2 \mathbf{I}_2 \bullet_3 \mathbf{I}_1. \tag{5.1}$$

However, it cannot be decomposed in two rank-$(2, 2, 2)$ terms. We prove this by contradiction. Assume that a decomposition in two rank-$(2, 2, 2)$ terms does exist:

$$\mathcal{T} = \mathcal{D}_1 \bullet_1 \mathbf{A}_1 \bullet_2 \mathbf{B}_1 \bullet_3 \mathbf{C}_1 + \mathcal{D}_2 \bullet_1 \mathbf{A}_2 \bullet_2 \mathbf{B}_2 \bullet_3 \mathbf{C}_2. \tag{5.2}$$

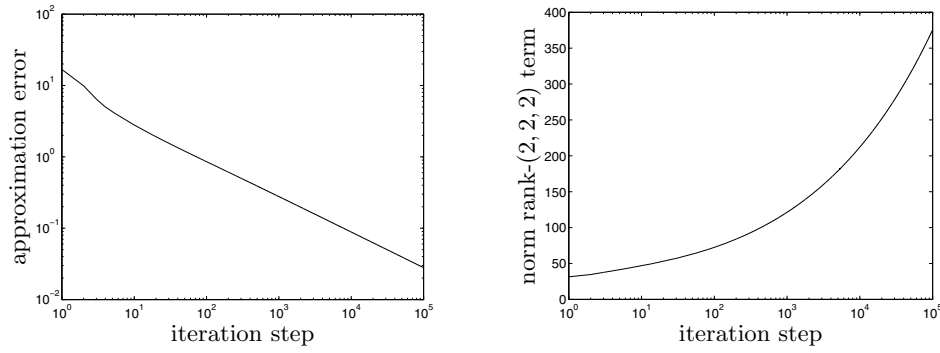


FIG. 5.1. *Visualization of the degeneracy in Example* 1. *Left*: *evolution of the approximation error. Right*: *evolution of the norm of the rank-*$(2,2,2)$ *terms.*

We can normalize this decomposition such that the first row of $\mathbf{C} = [\mathbf{C}_1\ \mathbf{C}_2]$ is equal to (1 0 1 0), and $\mathcal{D}_1 \bullet_3 (1\ 0) = \mathcal{D}_2 \bullet_3 (1\ 0) = \mathbf{I}_{2\times 2}$. Define $\mathbf{A} = [\mathbf{A}_1\ \mathbf{A}_2]$ and $\mathbf{B} = [\mathbf{B}_1\ \mathbf{B}_2]$. We have $\mathbf{T}_{I\times J,1} = \mathbf{I}_{4\times 4} = \mathbf{A}\cdot\mathbf{B}^T$. Hence, $\mathbf{A}$ and $\mathbf{B}$ are nonsingular. Define $\mathbf{X} = [\mathbf{x}_1\ \ldots\ \mathbf{x}_4] = \mathbf{A}^{-1}$ and $\mathbf{Y} = [\mathbf{y}_1\ \ldots\ \mathbf{y}_4] = \mathbf{B}^{-1}$. From (5.2) we have that all the $(I\times J)$ slices of $\tilde{\mathcal{T}} = \mathcal{T} \bullet_1 \mathbf{X} \bullet_2 \mathbf{Y}$ are block-diagonal, consisting of two (2×2) blocks. From the definition of $\mathcal{T}$, we have that $\tilde{\mathbf{T}}_{I\times J,4} = \mathbf{x}_1\cdot\mathbf{y}_4^T$. From the block-diagonality of this rank-1 matrix follows that, without loss of generality, we can assume that the third and fourth entries of $\mathbf{x}_1$ and $\mathbf{y}_4$ are zero. Further, we have that $\tilde{\mathbf{T}}_{I\times J,3} = \mathbf{x}_1\cdot\mathbf{y}_3^T + \mathbf{x}_2\cdot\mathbf{y}_4^T$. From the block-diagonality of this rank-2 matrix and the structure of $\mathbf{x}_1$ and $\mathbf{y}_4$ follows that the third and fourth entries of $\mathbf{x}_2$ and $\mathbf{y}_3$ are zero. Finally, we have that $\tilde{\mathbf{T}}_{I\times J,2} = \mathbf{x}_1\cdot\mathbf{y}_2^T + \mathbf{x}_3\cdot\mathbf{y}_4^T$. From the block-diagonality of this rank-2 matrix and the structure of $\mathbf{x}_1$ and $\mathbf{y}_4$ follows that the third and fourth entries of $\mathbf{x}_3$ and $\mathbf{y}_2$ are zero. We have a contradiction with the fact that $\mathbf{X}$ and $\mathbf{Y}$ are full rank. We conclude that $\mathcal{T}$ cannot be decomposed in a sum of two rank-$(2,2,2)$ terms.

On the other hand, there does not exist an approximation $\hat{\mathcal{T}}$, consisting of a sum of two rank-$(2,2,2)$ terms, that is optimal in the sense of minimizing the error $\|\mathcal{T} - \hat{\mathcal{T}}\|$. Define $\hat{\mathcal{T}}_n$ as follows, for increasing integer values of n:

$$\hat{\mathcal{T}}_n = \mathcal{E} \bullet_1 \mathbf{I}_1 \bullet_2 (\mathbf{I}_1 - n\mathbf{I}_2) \bullet_3 \mathbf{I}_1 + \mathcal{E} \bullet_1 \left(\mathbf{I}_1 + \frac{1}{n}\mathbf{I}_2\right) \bullet_2 (n\mathbf{I}_2) \bullet_3 \left(\mathbf{I}_1 + \frac{1}{n}\mathbf{I}_2\right). \tag{5.3}$$

We have

$$\hat{\mathcal{T}}_n = \mathcal{T} + \frac{1}{n}\,\mathcal{E} \bullet_1 \mathbf{I}_2 \bullet_2 \mathbf{I}_2 \bullet_3 \mathbf{I}_2.$$

Clearly, $\|\mathcal{T} - \hat{\mathcal{T}}_n\|$ goes to zero as n goes to infinity. However, at the same time the norms of the individual terms in (5.3) go to infinity. This shows that degeneracy also exists for block term decompositions.

Example 1. Figure 5.1 shows a typical degeneracy. We constructed a tensor $\mathcal{T}$ as in (5.1) with $\mathcal{E}$, however, defined by

$$e_{111} = -14 \quad e_{121} = -4 \quad e_{211} = 6 \quad e_{221} = 7,$$
$$e_{112} = 8 \quad e_{122} = 13 \quad e_{212} = 7 \quad e_{222} = 7.$$

The eigenvalues of $\mathbf{E}_{I\times J,1}\cdot\mathbf{E}_{I\times J,2}^{-1}$ are complex, so $\mathcal{E}$ is rank-3 in $\mathbb{R}$. The algorithm in Table 3.1 was used to approximate $\mathcal{T}$ by a sum of two rank-$(2,2,2)$ terms. The

left plot shows a monotonous decrease of the approximation error. The right plot shows the evolution of the norm of the rank-$(2, 2, 2)$ terms (the curves for both terms coincide).

6. Conclusion. We have derived ALS algorithms for the different block term decompositions that were introduced in [11]. ALS is actually a very simple approach. For PARAFAC, combining ALS with (exact) line search improves the performance [33]. An other technique that has proved useful for PARAFAC is the Levenberg–Marquardt type optimization [39]. When the tensor is tall in one mode, PARAFAC may often be computed by means of a simultaneous matrix decomposition [10]. Since the submission of this manuscript, we have been studying generalizations of such methods to block term decompositions [28, 29, 30, 31].

Acknowledgment. The authors wish to thank A. Stegeman (Heijmans Institute, The Netherlands) for proofreading an early version of the manuscript. A large part of this research was carried out when L. De Lathauwer and D. Nion were with the ETIS lab of the French Centre National de la Recherche Scientifique (C.N.R.S.).

REFERENCES

[1] G. Boutry, M. Elad, G.H. Golub, and P. Milanfar, *The generalized eigenvalue problem for nonsquare pencils using a minimal perturbation approach*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 582–601.

[2] D. Burdick, X. Tu, L. McGown, and D. Millican, *Resolution of multicomponent fluorescent mixtures by analysis of the excitation-emission-frequency array*, J. Chemometrics, 4 (1990), pp. 15–28.

[3] J. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 9 (1970), pp. 267–283.

[4] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, *Symmetric tensors and symmetric tensor rank*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1254–1279.

[5] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*, Ph.D. thesis, K.U.Leuven, Belgium, 1997.

[6] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[7] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[8] L. De Lathauwer and J. Vandewalle, *Dimensionality reduction in higher-order signal processing and rank-$(R_1, R_2, \ldots, R_N)$ reduction in multilinear algebra*, Linear Algebra Appl., 391 (2004), pp. 31–55.

[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Computation of the Canonical Decomposition by means of a simultaneous generalized Schur decompositition*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 295–327.

[10] L. De Lathauwer, *A link between the Canonical Decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666.

[11] L. De Lathauwer, *Decompositions of a higher-order tensor in block terms—Part* II: *Definitions and uniqueness*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1033–1066.

[12] V. de Silva and L.-H. Lim, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1084–1127.

[13] M. Elad, P. Milanfar, and G.H. Golub, *Shape from moments—an estimation theory perspective*, IEEE Trans. Signal Process., 52 (2004), pp. 1814–1829.

[14] L. Eldén and B. Savas, *A Newton–Grassmann Method for Computing the Best Multi-Linear Rank-(r_1, r_2, r_3) Approximation of a Tensor*, Tech. report LITH-MAT-R-2007-6-SE, Department of Mathematics, Linköping University, 2007.

[15] R.A. Harshman, *Foundations of the PARAFAC Procedure*: *Model and Conditions for an "Explanatory" Multi-Mode Factor Analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[16] R.A. Harshman and M.E. Lundy, *Data preprocessing and the extended Parafac model*, in Research Methods for Multimode Data Analysis, H.G. Law, C.W. Snyder, J.A. Hattie, and R.P. McDonald, eds., Praeger, New York, 1984, pp. 216–284.
[17] M. Ishteva, L. De Lathauwer, P.-A. Absil, and S. Van Huffel, *Dimensionality reduction for higher-order tensors: algorithms and applications*, Int. J. Pure Appl. Math., 42 (2008), pp. 337–343.
[18] J. Ja'Ja', *Optimal evaluation of bilinear forms*, SIAM J. Comput., 8 (1979), pp. 443–462.
[19] H. Kiers, *Towards a standardized notation and terminology in multiway analysis*, J. Chemometrics, 14 (2000), pp. 105–122.
[20] E. Kofidis and P.A. Regalia, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 863–884.
[21] P.M. Kroonenberg and J. de Leeuw, *Principal component analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 45 (1980), pp. 69–97.
[22] P.M. Kroonenberg, *Three-mode principal component analysis: illustrated with an example from attachment theory*, in Research Methods for Multimode Data Analysis, H.G. Law, C.W. Snyder, J.A. Hattie, and R.P. McDonald, eds., Praeger, New York, 1984, pp. 64–103.
[23] P.M. Kroonenberg, *Applied Multiway Data Analysis*, Wiley, New York, 2008.
[24] J.B. Kruskal, *Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.
[25] J.B. Kruskal, R.A. Harshman, and M.E. Lundy, *How 3-MFA data can cause degenerate PARAFAC solutions, among other relationships*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North–Holland, Amsterdam, 1989, pp. 115–122.
[26] S.E. Leurgans, R.T. Ross, and R.B. Abel, *A decomposition for three-way arrays*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1064–1083.
[27] B.C. Mitchell and D.S. Burdick, *Slowly converging Parafac sequences: Swamps and two-factor degeneracies*, J. Chem., 8 (1994), pp. 155–168.
[28] D. Nion and L. De Lathauwer, *Levenberg-Marquardt computation of the block factor model for blind multi-user access in wireless communications*, Proceedings of the 14th European Signal Processing Conference (Eusipco 2006), Florence, Italy, 2006.
[29] D. Nion and L. De Lathauwer, *A tensor-based blind DS-CDMA receiver using simultaneous matrix diagonalization*, Proceedings of the VIII IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2007), Helsinki, Finland, 2007.
[30] D. Nion and L. De Lathauwer, *Block component model based blind DS-CDMA receivers*, IEEE Trans. Signal. Process., to appear.
[31] D. Nion and L. De Lathauwer, *An enhanced line search scheme for complex-valued tensor decompositions. Application in DS-CDMA.*, Signal Process., 88 (2008), pp. 749–755.
[32] P. Paatero, *The multilinear engine—A table-driven, least squares program for solving multilinear problems, including the n-way parallel factor analysis model*, J. Comput. Graphical Statist., 8 (1999), pp. 854–888.
[33] M. Rajih, P. Comon, and R.A. Harshman, *Enhanced line search: a novel method to accelerate parafac*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1128–1147.
[34] C.R. Rao and S.K. Mitra, *Generalized Inverse of Matrices and Its Applications*, Wiley, New York, 1971.
[35] E. Sanchez and B.R. Kowalski, *Tensorial resolution: A direct trilinear decomposition*, J. Chemometrics, 4 (1990), pp. 29–45.
[36] R. Sands and F. Young, *Component models for three-way data: An alternating least squares algorithm with optimal scaling features*, Psychometrika, 45 (1980), pp. 39–67.
[37] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis. Applications in the Chemical Sciences*, Wiley, Chichester, UK, 2004.
[38] A. Stegeman, *Low-rank approximation of generic $p \times q \times 2$ arrays and diverging components in the Candecomp/Parafac model*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 988–1007.
[39] G. Tomasi and R. Bro, *A comparison of algorithms for fitting the PARAFAC model*, Comp. Stat. Data Anal., 50 (2006), pp. 1700–1734.
[40] L.R. Tucker, *The extension of factor analysis to three-dimensional matrices*, in Contributions to Mathematical Psychology, H. Gulliksen and N. Frederiksen, eds., Holt, Rinehart & Winston, New York, 1964, pp. 109–127.
[41] L.R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
[42] S.A. Vorobyov, Y. Rong, N.D. Sidiropoulos, and A.B. Gershman, *Robust iterative fitting of multilinear models*, IEEE Trans. Signal Process., 53 (2005), pp. 2678–2689.

[43] J. Weesie and H. Van Houwelingen, *GEPCAM Users' Manual: Generalized Principal Components Analysis with Missing Values*, Tech. report, Institute of Mathematical Statistics, University of Utrecht, Netherlands, 1983.

[44] T. Zhang and G.H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1084–1127

TENSOR RANK AND THE ILL-POSEDNESS OF THE BEST LOW-RANK APPROXIMATION PROBLEM*

VIN DE SILVA† AND LEK-HENG LIM‡

Abstract. There has been continued interest in seeking a theorem describing optimal low-rank approximations to tensors of order 3 or higher that parallels the Eckart–Young theorem for matrices. In this paper, we argue that the naive approach to this problem is doomed to failure because, unlike matrices, tensors of order 3 or higher can fail to have best rank-r approximations. The phenomenon is much more widespread than one might suspect: examples of this failure can be constructed over a wide range of dimensions, orders, and ranks, regardless of the choice of norm (or even Brègman divergence). Moreover, we show that in many instances these counterexamples have positive volume: they cannot be regarded as isolated phenomena. In one extreme case, we exhibit a tensor space in which *no* rank-3 tensor has an optimal rank-2 approximation. The notable exceptions to this misbehavior are rank-1 tensors and order-2 tensors (i.e., matrices). In a more positive spirit, we propose a natural way of overcoming the ill-posedness of the low-rank approximation problem, by using *weak solutions* when true solutions do not exist. For this to work, it is necessary to characterize the set of weak solutions, and we do this in the case of rank 2, order 3 (in arbitrary dimensions). In our work we emphasize the importance of closely studying concrete low-dimensional examples as a first step toward more general results. To this end, we present a detailed analysis of equivalence classes of $2 \times 2 \times 2$ tensors, and we develop methods for extending results upward to higher orders and dimensions. Finally, we link our work to existing studies of tensors from an algebraic geometric point of view. The rank of a tensor can in theory be given a semialgebraic description; in other words, it can be determined by a system of polynomial inequalities. We study some of these polynomials in cases of interest to us; in particular, we make extensive use of the *hyperdeterminant* Δ on $\mathbb{R}^{2\times2\times2}$.

Key words. numerical multilinear algebra, tensors, multidimensional arrays, multiway arrays, tensor rank, tensor decompositions, low-rank tensor approximations, hyperdeterminants, Eckart–Young theorem, principal component analysis, PARAFAC, CANDECOMP, Brègman divergence of tensors

AMS subject classifications. 14P10, 15A03, 15A21, 15A69, 15A72, 49M27, 62H25, 68P01

DOI. 10.1137/06066518X

1. Introduction.

Given an order-k tensor $A \in \mathbb{R}^{d_1\times\cdots\times d_k}$, one is often required to find a *best rank-r approximation* to A—in other words, determine vectors $\mathbf{x}_i \in \mathbb{R}^{d_1}, \mathbf{y}_i \in \mathbb{R}^{d_2}, \ldots, \mathbf{z}_i \in \mathbb{R}^{d_k}$, $i = 1, \ldots, r$, which minimize

$$\|A - \mathbf{x}_1 \otimes \mathbf{y}_1 \otimes \cdots \otimes \mathbf{z}_1 - \cdots - \mathbf{x}_r \otimes \mathbf{y}_r \otimes \cdots \otimes \mathbf{z}_r\|;$$

or, in short,

$$(\text{APPROX}(A, r)) \qquad \operatorname{argmin}_{\operatorname{rank}_\otimes(B)\le r} \|A - B\|.$$

Here $\|\cdot\|$ denotes some choice of norm on $\mathbb{R}^{d_1\times\cdots\times d_k}$. When $k = 2$, the problem is completely resolved for unitarily invariant norms on $\mathbb{R}^{m\times n}$ with the *Eckart–Young*

*Received by the editors July 16, 2006; accepted for publication (in revised form) by L. De Lathauwer June 7, 2007; published electronically September 25, 2008. This work was partially supported by DARPA grant 32905 and NSF grant DMS 01-01364.
http://www.siam.org/journals/simax/30-3/66518.html
†Department of Mathematics, Pomona College, Claremont, CA 91711-4411 (vin.desilva@pomona.edu).
‡Corresponding author. Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305-9025 (lekheng@stanford.edu). This author's research was supported by the Gerald J. Lieberman Fellowship from Stanford University.

theorem [28], which states that if

$$A = U\Sigma V = \sum\nolimits_{i=1}^{\operatorname{rank}(A)} \sigma_i \mathbf{u}_i \otimes \mathbf{v}_i, \quad \sigma_i \geq \sigma_{i+1},$$

is the singular value decomposition of $A \in \mathbb{R}^{m\times n}$, then a best rank-$r$ approximation is given by the first r terms in the above sum [33]. The best rank-r approximation problem for higher order tensors is a problem of central importance in the statistical analysis of multiway data [11, 16, 20, 21, 45, 50, 38, 56, 66, 67, 75, 76].

It is therefore not surprising that there has been continued interest in finding a satisfactory "singular value decomposition" and an "Eckart–Young theorem"-like result for tensors of higher order. The view expressed in the conclusion of [46] is representative of such efforts, and we reproduce it here:

> *An Eckart–Young type of best rank-r approximation theorem for tensors continues to elude our investigations but can perhaps eventually be attained by using a different norm or yet other definitions of orthogonality and rank.*

It will perhaps come as a surprise to the reader that the problem of finding an "Eckart–Young-type theorem" is ill-founded because of a more fundamental difficulty: the best rank-r approximation problem $\textsc{approx}(A, r)$ has no solution in general! This paper seeks to provide an answer to this and several related questions.

1.1. Summary. Since this is a long paper, we present an "executive summary" of selected results in this section and the next. We begin with the five main objectives of this article:

1. $\textsc{approx}(A, r)$ *is ill-posed for many* r. We will show that, regardless of the choice of norm, the problem of determining a best rank-r approximation for an order-k tensor in $\mathbb{R}^{d_1\times\cdots\times d_k}$ has no solution in general for $r = 2, \ldots, \min\{d_1, \ldots, d_k\}$ and $k \geq 3$. In other words, the best low-rank approximation problem for tensors is ill-posed for all orders (higher than 2), all norms, and many ranks.
2. $\textsc{approx}(A, r)$ *is ill-posed for many* A. We will show that the set of tensors that fail to have a best low-rank approximation has positive volume. In other words, such failures are not rare; if one randomly picks a tensor A in a suitable tensor space, then there is a nonzero probability that A will fail to have a best rank-r approximation for some $r < \operatorname{rank}_{\otimes}(A)$.
3. *Weak solutions to* $\textsc{approx}(A, r)$. We will propose a natural way to overcome the ill-posedness of the best rank-r approximation problem with the introduction of "weak solutions," which we explicitly characterize in the case $r = 2$, $k = 3$.
4. *Semialgebraic description of tensor rank.* From the Tarski–Seidenberg theorem in model theory [72, 65] we will deduce the following: for any $d_1, \ldots, d_k$, there exists a finite number of polynomial functions, $\Delta_1, \ldots, \Delta_m$, defined on $\mathbb{R}^{d_1\times\cdots\times d_k}$ such that the rank of any $A \in \mathbb{R}^{d_1\times\cdots\times d_k}$ is completely determined by the signs of $\Delta_1(A), \ldots, \Delta_m(A)$. We work this out in the special case $\mathbb{R}^{2\times2\times2}$.
5. *Reduction.* We will give techniques for reducing certain questions about tensors (orbits, invariants, limits) from high-dimensional tensor spaces to lower-dimensional tensor spaces. For instance, if two tensors in $\mathbb{R}^{c_1\times\cdots\times c_k}$ lie in distinct $\mathrm{GL}_{c_1,\ldots,c_k}(\mathbb{R})$-orbits, then they lie in distinct $\mathrm{GL}_{d_1,\ldots,d_k}(\mathbb{R})$-orbits in $\mathbb{R}^{d_1\times\cdots\times d_k}$ for any $d_i \geq c_i$.

The first objective is formally stated and proved in Theorem 4.10. The two notable exceptions where APPROX(A, r) has a solution are the cases $r = 1$ (approximation by rank-1 tensors) and $k = 2$ (A is a matrix). The standard way to prove these assertions is to use brute force: show that the sets where the approximators are to be found may be defined by polynomial equations. We will provide alternative elementary proofs of these results in Propositions 4.2 and 4.3 (see also Proposition 4.4).

The second objective is proved in Theorem 8.4, which holds true on $\mathbb{R}^{d_1 \times d_2 \times d_3}$ for arbitrary $d_1, d_2, d_3 \geq 2$. Stronger results can hold in specific cases: in Theorem 8.1, we will give an instance where *every* rank-r tensor fails to have a best rank-$(r-1)$ approximator.

The third objective is primarily possible because of the following theorem, which asserts that the boundary of the set of rank-2 tensors can be explicitly parameterized. The proof, and a discussion of weak solutions, is given in section 5.

THEOREM 1.1. *Let $d_1, d_2, d_3 \geq 2$. Let $A_n \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ be a sequence of tensors with $\operatorname{rank}_\otimes(A_n) \leq 2$ and*

$$\lim_{n \to \infty} A_n = A,$$

where the limit is taken in any norm topology. If the limiting tensor A has rank higher than 2, then $\operatorname{rank}_\otimes(A)$ must be exactly 3, and there exist pairs of linearly independent vectors $\mathbf{x}_1, \mathbf{y}_1 \in \mathbb{R}^{d_1}$, $\mathbf{x}_2, \mathbf{y}_2 \in \mathbb{R}^{d_2}$, $\mathbf{x}_3, \mathbf{y}_3 \in \mathbb{R}^{d_3}$ such that

$$A = \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{y}_3 + \mathbf{x}_1 \otimes \mathbf{y}_2 \otimes \mathbf{x}_3 + \mathbf{y}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3. \tag{1.1}$$

Furthermore, the above result is not vacuous since

$$A_n = n\left(\mathbf{x}_1 + \frac{1}{n}\mathbf{y}_1\right) \otimes \left(\mathbf{x}_2 + \frac{1}{n}\mathbf{y}_2\right) \otimes \left(\mathbf{x}_3 + \frac{1}{n}\mathbf{y}_3\right) - n\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3$$

is an example of a sequence that converges to A.

A few conclusions can immediately be drawn from Theorem 1.1: (i) the boundary points of all order-3 rank-2 tensors can be completely parameterized by (1.1); (ii) a sequence of order-3 rank-2 tensors cannot "jump rank" by more than 1; (iii) A in (1.1), in particular, is an example of a tensor that has no best rank-2 approximation.

The formal statements and proofs of the fourth objective appear in section 6. The fifth objective is exemplified by our approach throughout the paper; some specific technical tools are discussed in sections 5.1 and 7.5.

On top of these five objectives, we pick up the following smaller results along the way. Some of these results address frequently asked questions in tensor approximation. They are discussed in sections 4.3–4.7, respectively.

6. *Divergence of coefficients.* Whenever a low-rank sequence of tensors converges to a higher-rank tensor, some of the terms in the sequence must blow up. In examples of minimal rank, all the terms blow up.
7. *Maximum rank.* For $k \geq 3$, the maximum rank of an order-k tensor in $\mathbb{R}^{d_1 \times \cdots \times d_k}$ (where $d_i \geq 2$) always exceeds $\min(d_1, \ldots, d_k)$. In contrast, for matrices $\min(d_1, d_2)$ does bound the rank.
8. *Tensor rank can leap large gaps.* Conclusion (ii) above does not generalize to rank $r > 2$. We will show that a sequence of fixed rank tensors can converge to a limiting tensor of arbitrarily higher rank.
9. *Brègman divergences do not help.* If we replace the norm by any continuous measure of "nearness" (including nonmetric measures like Brègman divergences), it does not change the ill-foundedness of APPROX(A, r).

10. *Leibniz tensors.* We will construct a rich family of sequences of tensors with degenerate limits, labeled by partial derivative operators. The special case $L_3(1)$ is in fact the principal example (1.1) in this paper.

1.2. Relation to prior work. The existence of tensors that can fail to have a best rank-r approximation has been known to algebraic geometers as early as the 19th century, albeit in a different language—the locus of rth secant planes to a Segre variety may not define a (closed) algebraic variety. It is also known to computational complexity theorists as the phenomenon underlying the concept of *border rank* [5, 6, 12, 48, 54] and is related to (but different from) what chemometricians and psychometricians call "CANDECOMP/PARAFAC degeneracy" [49, 51, 63, 68, 69]. We do not claim to be the first to have found such an example—that honor belongs to Bini, Capovani, Lotti, and Romani, who gave an explicit example of a sequence of rank-5 tensors converging to a rank-6 tensor in 1979 [7]. The novelty of Theorem 1.1 is not in demonstrating that a tensor may be approximated arbitrarily well by tensors of strictly lower rank but in *characterizing all such tensors* in the order-3 rank-2 case.

Having said this, we would like to point out that the ill-posedness of the best rank-r approximation problem for high-order tensors is not at all well known, as is evident from the paragraph quoted earlier as well as other discussions in recent publications [44, 45, 46, 47, 80]. One likely reason is that in algebraic geometry, computational complexity, chemometrics, and psychometrics, the problem is neither stated in the form nor viewed in the light of obtaining a best low-rank approximation with respect to a choice of norm (we give several equivalent formulations of $\text{APPROX}(A, r)$ in Proposition 4.1). As such, one goal of this paper will be to debunk, once and for all, the question of finding best low-rank approximations for tensors of order 3 or higher. As we stated earlier (as our first and second objectives), our contribution will be to show that such failures (i) can and will occur for tensors of *any* order higher than 2, (ii) will occur for tensors of many different ranks, (iii) will occur regardless of the choice of norm, and (iv) will occur with nonzero probability. Formally, we have the following two theorems (which will appear as Theorems 4.10 and 8.4 subsequently).

THEOREM 1.2. *Let $k \geq 3$ and $d_1, \dots, d_k \geq 2$. For any s such that $2 \leq s \leq \min\{d_1, \dots, d_k\}$, there exists $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ with $\operatorname{rank}_{\otimes}(A) = s$ such that A has no best rank-r approximation for some $r < s$. The result is independent of the choice of norms.*

THEOREM 1.3. *If $d_1, d_2, d_3 \geq 2$, then the set*

$$\{A \in \mathbb{R}^{d_1 \times d_2 \times d_3} \mid A \text{ does not have a best rank-2 approximation}\}$$

has positive volume; indeed, it contains a nonempty open set.

A few features distinguish our work in this paper from existing studies in algebraic geometry [13, 14, 54, 55, 79] and algebraic computational complexity [2, 3, 5, 6, 7, 8, 12, 70]: (i) we are interested in tensors over $\mathbb{R}$ as opposed to tensors over $\mathbb{C}$ (it is well known that the rank of a tensor is dependent on the underlying field; cf. (7.5) and [4]); (ii) our interest is not limited to order-3 tensors (as is often the case in algebraic computational complexity)—we would like to prove results that hold for tensors of any order $k \geq 3$; (iii) since we are interested in questions pertaining to approximations in the norm, the Euclidean (norm-induced) topology will be more relevant than the

Zariski topology[1] on the tensor product spaces—note in particular that the claim that a set is not closed in the Euclidean topology is a stronger statement than the corresponding claim in Zariski topology.

Our work in this paper in general, and in section 4.2 in particular, is related to studies of "CANDECOMP/PARAFAC degeneracy" or "diverging CANDECOMP/PARAFAC components" in psychometrics and chemometrics [49, 51, 63, 68, 69]. Diverging coefficients are a necessary consequence of the ill-posedness of APPROX(A, r) (see Propositions 4.8 and 4.9). In fact, examples of "k-factor divergence" abound for arbitrary k—see sections 4.4 and 4.7 for various constructions.

Section 5.4 discusses how the nonexistence of a best rank-r approximation poses serious difficulties for multilinear statistical models based on such approximations. In particular, we will see (i) why it is meaningless to ask for a "good" rank-r approximation when a best rank-r approximation does not exist; (ii) why even a small perturbation to a rank-r tensor can result in a tensor that has no best rank-r approximation; and (iii) why the computational feasibility of finding a "good" rank-r approximation is questionable.

1.3. Outline of the paper. Section 2 introduces the basic algebra of tensors and k-way arrays. Section 3 defines tensor rank and gives some of its known (and unknown) algebraic properties. Section 4 studies the topological properties of tensor rank and the phenomenon of rank-jumping. Section 5 characterizes the problematic tensors in $\mathbb{R}^{2\times2\times2}$ and discusses the implications for approximation problems. Section 6 gives a short exposition of the semialgebraic point of view. Section 7 classifies tensors in $\mathbb{R}^{2\times2\times2}$ by orbit type. The orbit structure of tensor spaces is studied from several different aspects. Section 8 is devoted to the result that failure of APPROX$(A, 2)$ occurs on a set of positive volume.

2. Tensors. Even though tensors are well-studied objects in the standard graduate mathematics curriculum [1, 27, 41, 52, 64] and more specifically in multilinear algebra [9, 34, 59, 60, 62, 78], a "tensor" continues to be viewed as a mysterious object by outsiders. We feel that we should say a few words to demystify the term.

In mathematics, the question "What is a vector?" has the simple answer "A vector is an element of a vector space"—in other words, a vector is characterized by the axioms that define the algebraic operations on a vector space. In physics, however, the question "What is a vector?" often means "What kinds of physical quantities can be represented by vectors?" The criterion has to do with the change of basis theorem: an n-dimensional vector is an "object" that is represented by n real numbers once a basis is chosen *only* if those real numbers transform themselves as expected when one changes the basis. For exactly the same reason, the meaning of a tensor is obscured by its more restrictive use in physics. In physics (and also engineering), a tensor is an "object" represented by a k-way array of real numbers that transforms according to certain rules (cf. (2.2)) under a change of basis. In mathematics, these "transformation rules" are simply consequences of the multilinearity of the tensor product and the change of basis theorem for vectors. Today, books written primarily for a physics audience [32, 61] have increasingly adopted the mathematical definition, but a handful of recently published books continue to propagate the obsolete (and vague) definition. To add to the confusion, "tensor" is frequently used to refer to a

[1] Note that the Zariski topology on $\mathbb{k}^n$ is defined for any field $\mathbb{k}$ (not just algebraically closed ones). It is the weakest topology such that all polynomial functions are continuous. In particularly, the closed sets are precisely the zero sets of collections of polynomials.

tensor field (e.g., metric tensor, stress tensor, Riemann curvature tensor).

For our purposes, an order-k *tensor* $\mathbf{A}$ is simply an element of a *tensor product* of k real vector spaces, $V_1 \otimes V_2 \otimes \cdots \otimes V_k$, as defined in any standard algebra textbook [1, 9, 27, 34, 41, 52, 59, 60, 62, 64, 78]. Up to a choice of bases on $V_1, \dots, V_k$, such an element may be *coordinatized*, i.e., represented as a k-way array A of real numbers—much as an element of an n-dimensional vector space may be, up to a choice of basis, represented by an n-tuple of numbers in $\mathbb{R}^n$. We will let $\mathbb{R}^{d_1 \times \cdots \times d_k}$ denote the vector space of k-way array of real numbers $A = [\![a_{j_1 \cdots j_k}]\!]_{j_1=1,\dots,j_k=1}^{d_1,\dots,d_k}$ with addition and scalar multiplication defined coordinatewise:

$$[\![a_{j_1 \cdots j_k}]\!] + [\![b_{j_1 \cdots j_k}]\!] := [\![a_{j_1 \cdots j_k} + b_{j_1 \cdots j_k}]\!] \quad \text{and} \quad \lambda [\![a_{j_1 \cdots j_k}]\!] := [\![\lambda a_{j_1 \cdots j_k}]\!]. \tag{2.1}$$

A k-way array of numbers (or k-array) is also sometimes referred to as a k-dimensional *hypermatrix* [30].

It may be helpful to think of a k-array as a *data structure*, convenient for representing or storing the coefficients of a tensor with respect to a set of bases. The tensor itself carries with it an *algebraic structure*, by virtue of being an element of a tensor product of vector spaces. Once bases have been chosen for these vector spaces, we may view the order-k tensor as a k-way array equipped with the algebraic operations defined in (2.1) and (2.3). Despite this correspondence, it is not wise to regard "tensor" as being synonymous with "array."

Notation. We will denote elements of abstract tensor spaces in boldface uppercase letters, whereas k-arrays will be denoted in italic uppercase letters. Thus $\mathbf{A}$ is an abstract tensor, which may be represented by an array of numbers A with respect to a basis. We will use double brackets to enclose the entries of a k-array—$A = [\![a_{j_1 \cdots j_k}]\!]_{j_1=1,\dots,j_k=1}^{d_1,\dots,d_k}$—and when there is no risk of confusion, we will leave out the range of the indices and simply write $A = [\![a_{j_1 \cdots j_k}]\!]$.

2.1. Multilinear matrix multiplication. Matrices can act on other matrices through two independent multiplication operations: left-multiplication and right-multiplication. Matrices act on order-3 tensors via *three* different multiplication operations. These can be combined into a single formula. If $A = [\![a_{ijk}]\!] \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ and $L = [\lambda_{pi}] \in \mathbb{R}^{c_1 \times d_1}$, $M = [\mu_{qj}] \in \mathbb{R}^{c_2 \times d_2}$, $N = [\nu_{rk}] \in \mathbb{R}^{c_3 \times d_3}$, then the array A may be transformed into an array $A' = [\![a'_{pqr}]\!] \in \mathbb{R}^{c_1 \times c_2 \times c_3}$ by the equation

$$a'_{pqr} = \sum\nolimits_{i,j,k=1}^{d_1,d_2,d_3} \lambda_{pi} \mu_{qj} \nu_{rk} a_{ijk}. \tag{2.2}$$

We call this operation the *multilinear multiplication* of A by matrices L, M, N, which we write succinctly as

$$A' = (L, M, N) \cdot A.$$

Informally, we are multiplying the 3-way array A on its three "sides" by the matrices L, M, N, respectively.

Remark. This notation is standard in mathematics—the elements of a product $G_1 \times G_2 \times G_3$ are generally grouped in the form (L, M, N), and when a set with some algebraic structure G acts on another set X, the result of $g \in G$ acting on $x \in X$ is almost universally written $g \cdot x$ [1, 9, 27, 41, 52, 64]. Here we are just looking at the case when $G = \mathbb{R}^{c_1 \times d_1} \times \mathbb{R}^{c_2 \times d_2} \times \mathbb{R}^{c_3 \times d_3}$ and $X = \mathbb{R}^{d_1 \times d_2 \times d_3}$. This is consistent with notation adopted in earlier work [42], but more recent publications such as [20, 21] have used $A \times_1 L^\top \times_2 M^\top \times_3 N^\top$ in place of $(L, M, N) \cdot A$.

Multilinear matrix multiplication extends in a straightforward way to arrays of arbitrary order: if $A = [\![a_{i_1\cdots i_k}]\!] \in \mathbb{R}^{d_1\times\cdots\times d_k}$ and $L_1 = [\lambda_{ij}^{(1)}] \in \mathbb{R}^{c_1\times d_1}, \dots, L_k = [\lambda_{ij}^{(k)}] \in \mathbb{R}^{c_k\times d_k}$, then $A' = (L_1, \dots, L_k) \cdot A$ is the array $A' = [\![a'_{i_1\cdots i_k}]\!] \in \mathbb{R}^{c_1\times\cdots\times c_k}$ given by

$$a'_{i_1\cdots i_k} = \sum\nolimits_{i_1,\dots,i_k=1}^{d_1,\dots,d_k} \lambda_{i_1 j_1} \cdots \lambda_{i_k j_k} a_{j_1\cdots j_k}. \tag{2.3}$$

We will now see how a 3-way array representing a tensor in $V_1 \otimes V_2 \otimes V_3$ transforms under changes of bases of the vector spaces V_i. Suppose the 3-way array $A = [\![a_{ijk}]\!] \in \mathbb{R}^{d_1\times d_2\times d_3}$ represents an order-3 tensor $\mathbf{A} \in V_1 \otimes V_2 \otimes V_3$ with respect to bases $\mathcal{B}_1 = \{\mathbf{e}_i \mid i = 1, \dots, d_1\}$, $\mathcal{B}_2 = \{\mathbf{f}_j \mid j = 1, \dots, d_2\}$, $\mathcal{B}_3 = \{\mathbf{g}_k \mid k = 1, \dots, d_3\}$ on V_1, V_2, V_3, i.e.,

$$\mathbf{A} = \sum\nolimits_{i,j,k=1}^{d_1,d_2,d_3} a_{ijk} \mathbf{e}_i \otimes \mathbf{f}_j \otimes \mathbf{g}_k. \tag{2.4}$$

Suppose we choose different bases $\mathcal{B}'_1 = \{\mathbf{e}'_i \mid i = 1, \dots, d_1\}$, $\mathcal{B}'_2 = \{\mathbf{f}'_j \mid j = 1, \dots, d_2\}$, $\mathcal{B}'_3 = \{\mathbf{g}'_k \mid k = 1, \dots, d_3\}$ on V_1, V_2, V_3, where

$$\mathbf{e}_i = \sum\nolimits_{p=1}^{d_1} \lambda_{ip} \mathbf{e}'_p, \quad \mathbf{f}_j = \sum\nolimits_{q=1}^{d_2} \mu_{jq} \mathbf{f}'_q, \quad \mathbf{g}_k = \sum\nolimits_{r=1}^{d_3} \nu_{kr} \mathbf{g}'_r, \tag{2.5}$$

and $L = [\lambda_{pi}] \in \mathbb{R}^{d_1\times d_1}$, $M = [\mu_{qj}] \in \mathbb{R}^{d_2\times d_2}$, $N = [\nu_{rk}] \in \mathbb{R}^{d_3\times d_3}$ are the respective change-of-basis matrices. Substituting the expressions for (2.5) into (2.4), we get

$$\mathbf{A} = \sum\nolimits_{p,q,r=1}^{d_1,d_2,d_3} a'_{pqr} \mathbf{e}'_p \otimes \mathbf{f}'_q \otimes \mathbf{g}'_r,$$

where

$$a'_{pqr} = \sum\nolimits_{i,j,k=1}^{d_1,d_2,d_3} \lambda_{pi} \mu_{qj} \nu_{rk} a_{ijk} \tag{2.6}$$

or, more simply, $A' = (L, M, N) \cdot A$. Here the 3-way array $A' = [\![a'_{pqr}]\!] \in \mathbb{R}^{d_1\times d_2\times d_3}$ represents $\mathbf{A}$ with respect to the new choice of bases $\mathcal{B}'_1, \mathcal{B}'_2, \mathcal{B}'_3$.

All of this extends immediately to order-k tensors and k-way arrays. Henceforth, when a choice of basis is implicit, we will not distinguish between an order-k tensor and the k-way array that represents it.

The change-of-basis matrices L, M, N in the discussion above are of course invertible; in other words they belong to their respective *general linear* groups. We write $\mathrm{GL}_d(\mathbb{R})$ for the group of nonsingular matrices in $\mathbb{R}^{d\times d}$. Thus $L \in \mathrm{GL}_{d_1}(\mathbb{R})$, $M \in \mathrm{GL}_{d_2}(\mathbb{R})$, and $N \in \mathrm{GL}_{d_3}(\mathbb{R})$. In addition to general linear transformations, it is natural to consider orthogonal transformations. We write $\mathrm{O}_d(\mathbb{R})$ for the subgroup of $\mathrm{GL}_d(\mathbb{R})$ of transformations which preserve the Euclidean inner product. The following shorthand is helpful:

$$\mathrm{GL}_{d_1,\dots,d_k}(\mathbb{R}) := \mathrm{GL}_{d_1}(\mathbb{R}) \times \cdots \times \mathrm{GL}_{d_k}(\mathbb{R}),$$
$$\mathrm{O}_{d_1,\dots,d_k}(\mathbb{R}) := \mathrm{O}_{d_1}(\mathbb{R}) \times \cdots \times \mathrm{O}_{d_k}(\mathbb{R}).$$

Then $\mathrm{O}_{d_1,\dots,d_k}(\mathbb{R}) \le \mathrm{GL}_{d_1,\dots,d_k}(\mathbb{R})$, and both groups act on $\mathbb{R}^{d_1\times\cdots\times d_k}$ via multilinear multiplication.

DEFINITION 2.1. *Two tensors $A, A' \in \mathbb{R}^{d_1\times\cdots\times d_k}$ are said to be* GL*-equivalent (or simply "equivalent") if there exists $(L_1, \dots, L_k) \in \mathrm{GL}_{d_1,\dots,d_k}(\mathbb{R})$ such that $A' =$*

$(L_1, \dots, L_k) \cdot A$. *More strongly, we say that* A, A' *are* O*-equivalent if such a transformation* L *can be found in* $\mathrm{O}_{d_1,\dots,d_k}(\mathbb{R})$.

For example, if $V_1, \dots, V_k$ are vector spaces and $\dim(V_i) = d_i$, then $A, A' \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ represent the same tensor in $V_1 \otimes \cdots \otimes V_k$ with respect to two different bases if and only if A, A' are GL-equivalent.

We finish with some trivial properties of multilinear matrix multiplication: for $A, B \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ and $\alpha, \beta \in \mathbb{R}$,

$$(L_1, \dots, L_k) \cdot (\alpha A + \beta B) = \alpha (L_1, \dots, L_k) \cdot A + \beta (L_1, \dots, L_k) \cdot B \tag{2.7}$$

and for $L_i \in \mathbb{R}^{c_i \times d_i}$, $M_i \in \mathbb{R}^{b_i \times c_i}$, $i = 1, \dots, k$,

$$(M_1, \dots, M_k) \cdot [(L_1, \dots, L_k) \cdot A] = (M_1 L_1, \dots, M_k L_k) \cdot A. \tag{2.8}$$

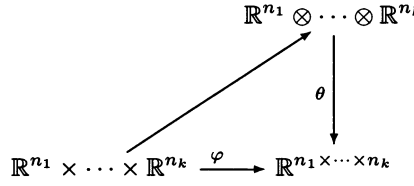
Last, the name *multilinear* matrix multiplication is justified since for any $M_i, N_i \in \mathbb{R}^{c_i \times d_i}$, $\alpha, \beta \in \mathbb{R}$,

$$\begin{aligned}(L_1, \dots, \alpha M_i + \beta N_i, \dots, L_k) \cdot A = \alpha (L_1, \dots, M_i, \dots, L_k) \cdot A \\ + \beta (L_1, \dots, N_i, \dots, L_k) \cdot A.\end{aligned} \tag{2.9}$$

2.2. Outer-product rank and outer-product decomposition of a tensor. Let $\mathbb{R}^{d_1} \otimes \cdots \otimes \mathbb{R}^{d_k}$ be the tensor product of the vector spaces $\mathbb{R}^{d_1}, \dots, \mathbb{R}^{d_k}$. Note that the *Segre map*

$$\mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_k} \to \mathbb{R}^{d_1 \times \cdots \times d_k}, \qquad (\mathbf{x}_1, \dots, \mathbf{x}_k) \mapsto [\![x^{(1)}_{j_1} \cdots x^{(k)}_{j_k}]\!]^{d_1,\dots,d_k}_{j_1,\dots,j_k=1} \tag{2.10}$$

is multilinear and so by the universal property of the tensor product [1, 9, 27, 34, 41, 52, 59, 60, 62, 64, 78], we have a unique linear map φ such that the following diagram commutes:



Clearly,

$$\varphi(\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_k) = [\![x^{(1)}_{j_1} \cdots x^{(k)}_{j_k}]\!]^{d_1,\dots,d_k}_{j_1,\dots,j_k=1} \tag{2.11}$$

and φ is a vector space isomorphism since $\dim(\mathbb{R}^{d_1 \times \cdots \times d_k}) = \dim(\mathbb{R}^{d_1} \otimes \cdots \otimes \mathbb{R}^{d_k}) = d_1 \cdots d_k$. Henceforth we will not distinguish between these two spaces. The elements of $\mathbb{R}^{d_1} \otimes \cdots \otimes \mathbb{R}^{d_k} \cong \mathbb{R}^{d_1 \times \cdots \times d_k}$ will be called a tensor and we will also drop φ in (2.11) and write

$$\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_k = [\![x^{(1)}_{j_1} \cdots x^{(k)}_{j_k}]\!]^{d_1,\dots,d_k}_{j_1,\dots,j_k=1}. \tag{2.12}$$

Note that the symbol $\otimes$ in (2.11) denotes the formal tensor product and by dropping φ, we are using the same symbol $\otimes$ to define the *outer product* of the vectors $\mathbf{x}_1, \dots, \mathbf{x}_k$ via the formula (2.12). Hence, a tensor can be represented either as a k-dimensional array or as a sum of formal tensor products of k vectors, where the equivalence between

these two objects is established by taking the formal tensor product of k vectors as defining a k-way array via (2.12).

It is clear that the map in (2.10) is not surjective—the image consists precisely of the *decomposable tensors*: a tensor $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ is said to be *decomposable* if it can be written in the form

$$A = \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_k$$

with $\mathbf{x}_i \in \mathbb{R}^{d_i}$ for $i = 1, \ldots, k$. It is easy to see that multilinear matrix multiplication of decomposable tensors obeys the formula

$$(L_1, \ldots, L_k) \cdot (\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_k) = L_1\mathbf{x}_1 \otimes \cdots \otimes L_k\mathbf{x}_k. \tag{2.13}$$

Remark. The outer product can be viewed as a special case of multilinear matrix multiplication. For example, a linear combination of outer products of vectors may be expressed in terms of multilinear matrix multiplication:

$$\sum\nolimits_{i=1}^{r} \lambda_i \mathbf{x}_i \otimes \mathbf{y}_i \otimes \mathbf{z}_i = (X, Y, Z) \cdot \Lambda$$

with matrices $X = [\mathbf{x}_1, \ldots, \mathbf{x}_r] \in \mathbb{R}^{l \times r}$, $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_r] \in \mathbb{R}^{m \times r}$, $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_r] \in \mathbb{R}^{n \times r}$ and a "diagonal tensor" $\Lambda = \operatorname{diag}[\lambda_1, \ldots, \lambda_r] \in \mathbb{R}^{r \times r \times r}$.

We now come to the main concept of interest in this paper.

DEFINITION 2.2. *A tensor has* outer-product rank r *if it can be written as a sum of* r *decomposable tensors, but no fewer. We will write* $\operatorname{rank}_{\otimes}(A)$ *for the outer-product rank of* A. *So*

$$\operatorname{rank}_{\otimes}(A) := \min \left\{ r \mid A = \sum\nolimits_{i=1}^{r} \mathbf{u}_i \otimes \mathbf{v}_i \otimes \cdots \otimes \mathbf{z}_i \right\}.$$

Note that a nonzero decomposable tensor has outer-product rank 1.

Despite several claims of originality as well as many misplaced attributions to these claims, the concepts of tensor rank and the decomposition of a tensor into a sum of outer products of vectors was the product of much earlier work by Frank L. Hitchcock in 1927 [39, 40]. We call this the outer-product rank mainly to distinguish it from the *multilinear rank* to be defined in section 2.5 (also due to Hitchcock), but we will use the term *rank* or *tensor rank* most of the time when there is no danger of confusion.

LEMMA 2.3 (invariance of tensor rank). (1) *If* $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ *and* $(L_1, \ldots, L_k) \in \mathbb{R}^{c_1 \times d_1} \times \cdots \times \mathbb{R}^{c_k \times d_k}$, *then*

$$\operatorname{rank}_{\otimes}((L_1, \ldots, L_k) \cdot A) \leq \operatorname{rank}_{\otimes}(A). \tag{2.14}$$

(2) *If* $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ *and* $(L_1, \ldots, L_k) \in \mathrm{GL}_{d_1,\ldots,d_k}(\mathbb{R}) := \mathrm{GL}_{d_1}(\mathbb{R}) \times \cdots \times \mathrm{GL}_{d_k}(\mathbb{R})$, *then*

$$\operatorname{rank}_{\otimes}((L_1, \ldots, L_k) \cdot A) = \operatorname{rank}_{\otimes}(A). \tag{2.15}$$

Proof. Inequality (2.14) follows from (2.13) and (2.7). Indeed, if $A = \sum_{j=1}^{r} \mathbf{x}_1^j \otimes \cdots \otimes \mathbf{x}_k^j$, then $(L_1, \ldots, L_k) \cdot A = \sum_{j=1}^{r} L_1\mathbf{x}_1^j \otimes \cdots \otimes L_k\mathbf{x}_k^j$. Furthermore, if the L_i are invertible, then by (2.8) we get

$$A = (L_1^{-1}, \ldots, L_k^{-1}) \cdot [(L_1, \ldots, L_k) \cdot A],$$

and so $\operatorname{rank}_{\otimes}(A) \leq \operatorname{rank}_{\otimes}((L_1, \ldots, L_k) \cdot A)$, and hence (2.15). ∎

2.3. The outer product and direct sum operations on tensors. The outer product of vectors defined earlier is a special case of the *outer product* of two tensors. Let $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ be a tensor of order k and $B \in \mathbb{R}^{c_1 \times \cdots \times c_\ell}$ be a tensor of order ℓ; then the outer product of A and B is the tensor $C := A \otimes B \in \mathbb{R}^{d_1 \times \cdots \times d_k \times c_1 \times \cdots \times c_l}$ of order $k + \ell$ defined by

$$c_{i_1 \cdots i_k j_1 \cdots j_l} = a_{i_1 \cdots i_k} b_{j_1 \cdots j_l}.$$

The *direct sum* of two order-k tensors $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ and $B \in \mathbb{R}^{c_1 \times \cdots \times c_k}$ is the order-k tensor $C := A \oplus B \in \mathbb{R}^{(c_1+d_1) \times \cdots \times (c_k+d_k)}$ defined by

$$c_{i_1,\ldots,i_k} = \begin{cases} a_{i_1,\ldots,i_k} & \text{if } 1 \le i_\alpha \le d_\alpha,\ \alpha = 1,\ldots,k; \\ b_{i_1-d_1,\ldots,i_k-d_k} & \text{if } d_\alpha + 1 \le i_\alpha \le c_\alpha + d_\alpha,\ \alpha = 1,\ldots,k; \\ 0 & \text{otherwise.} \end{cases}$$

For matrices, the direct sum of $A \in \mathbb{R}^{m_1 \times n_1}$ and $B \in \mathbb{R}^{m_2 \times n_2}$ is simply the block-diagonal matrix

$$A \oplus B = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \in \mathbb{R}^{(m_1+m_2) \times (n_1+n_2)}.$$

The direct sum of two order-3 tensors $A \in \mathbb{R}^{l_1 \times m_1 \times n_1}$ and $B \in \mathbb{R}^{l_2 \times m_2 \times n_2}$ is a "block tensor" with A in the $(1,1,1)$-block and B in the $(2,2,2)$-block

$$A \oplus B = \left[\begin{array}{cc|cc} A & 0 & 0 & 0 \\ 0 & 0 & 0 & B \end{array}\right] \in \mathbb{R}^{(l_1+l_2) \times (m_1+m_2) \times (n_1+n_2)}.$$

In abstract terms, if U_i, V_i, W_i are vector spaces such that $W_i = U_i \oplus V_i$ for $i = 1,\ldots,k$, then tensors $A \in U_1 \otimes \cdots \otimes U_k$ and $B \in V_1 \otimes \cdots \otimes V_k$ have direct sum $A \oplus B \in W_1 \otimes \cdots \otimes W_k$.

2.4. Tensor subspaces. Whenever $c \le d$ there is a canonical embedding $\mathbb{R}^c \subseteq \mathbb{R}^d$ given by identifying the c coordinates of $\mathbb{R}^c$ with the first c coordinates of $\mathbb{R}^d$.

Let $c_i \le d_i$ for $i = 1,\ldots,k$. Then there is a canonical embedding $\mathbb{R}^{c_1 \times \cdots \times c_k} \subset \mathbb{R}^{d_1 \times \cdots \times d_k}$, defined as the tensor product of the embeddings $\mathbb{R}^{c_i} \subseteq \mathbb{R}^{d_i}$. We say that $\mathbb{R}^{c_1 \times \cdots \times c_k}$ is a *tensor subspace* of $\mathbb{R}^{d_1 \times \cdots \times d_k}$. More generally, if U_i, V_i are vector spaces with $U_i \subset V_i$ for $i = 1,\ldots,k$, then there is an inclusion $U_1 \otimes \cdots \otimes U_k \subset V_1 \otimes \cdots \otimes V_k$ defined as the tensor product of the inclusions $U_i \subset V_i$. Again we say that $U_1 \otimes \cdots \otimes U_k$ is a tensor subspace of $V_1 \otimes \cdots \otimes V_k$.

If $B \in \mathbb{R}^{c_1 \times \cdots \times c_k}$ then its image under the canonical embedding into $\mathbb{R}^{d_1 \times \ldots d_k}$ can be written in the form $B \oplus 0$, where $0 \in \mathbb{R}^{(d_1-c_1) \times \cdots \times (d_k-c_k)}$ is the zero tensor. A tensor $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ is said to be GL-*equivalent* (or simply "equivalent") to B if there exists $(L_1,\ldots,L_k) \in \mathrm{GL}_{d_1,\ldots,d_k}(\mathbb{R})$ such that $B \oplus 0 = (L_1,\ldots,L_k) \cdot A$. More strongly, we say that A is O-*equivalent* ("orthogonally equivalent") to B if such a transformation can be found in $\mathrm{O}_{d_1,\ldots,d_k}(\mathbb{R})$.

We note that A is GL-equivalent to B if and only if there exist full-rank matrices $M_i \in \mathbb{R}^{d_i \times c_i}$ such that $A = (M_1,\ldots,M_k) \cdot B$. In one direction, M_i can be obtained as the first c_i columns of L_i^{-1}. In the other direction, L_i^{-1} can be obtained from M_i by adjoining extra columns. There is a similar statement for O-equivalence. Instead of full rank, the condition is that the matrices M_i have orthogonal columns.

An important simplifying principle in tensor algebra is that questions about a tensor—such as "What is its rank?"—can sometimes, as we shall see, be reduced to analogous questions about an equivalent tensor in a lower-dimensional tensor subspace.

2.5. Multilinear rank and multilinear decomposition of a tensor. Although we focus on outer product rank in this paper, there is a simpler notion of multilinear rank which directly generalizes the column and row ranks of a matrix to higher order tensors.

For convenience, we will consider order-3 tensors only. Let $A = [\![a_{ijk}]\!] \in \mathbb{R}^{d_1 \times d_2 \times d_3}$. For fixed values of $j \in \{1, \dots, d_2\}$ and $k \in \{1, \dots, d_3\}$, consider the vector $A_{\bullet jk} := [a_{ijk}]_{i=1}^{d_1} \in \mathbb{R}^{d_1}$. Likewise consider (column) vectors $A_{i\bullet k} := [a_{ijk}]_{j=1}^{d_2} \in \mathbb{R}^{d_2}$ for fixed values of i, k, and consider (row) vectors $A_{ij\bullet} := [a_{ijk}]_{k=1}^{d_3} \in \mathbb{R}^{d_3}$ for fixed values of i, j. In analogy with row rank and column rank, define

$$\begin{aligned} r_1(A) &:= \dim(\operatorname{span}_{\mathbb{R}}\{A_{\bullet jk} \mid 1 \le j \le d_2, 1 \le k \le d_3\}), \\ r_2(A) &:= \dim(\operatorname{span}_{\mathbb{R}}\{A_{i\bullet k} \mid 1 \le i \le d_1, 1 \le k \le d_3\}), \\ r_3(A) &:= \dim(\operatorname{span}_{\mathbb{R}}\{A_{ij\bullet} \mid 1 \le i \le d_1, 1 \le j \le d_2\}). \end{aligned}$$

For another interpretation, note that $\mathbb{R}^{d_1 \times d_2 \times d_3}$ can be viewed as $\mathbb{R}^{d_1 \times d_2 d_3}$ by ignoring the multiplicative structure between the second and third factors. Then $r_1(A)$ is simply the rank of A regarded as $d_1 \times d_2 d_3$ matrix. There are similar definitions for $r_2(A)$ and $r_3(A)$.

The *multilinear rank* of A, denoted[2] $\operatorname{rank}_{\boxplus}(A)$, is the 3-tuple $(r_1(A), r_2(A), r_3(A))$. Again, this concept is not new but was first explored by Hitchcock in the same papers where he introduced tensor rank [39, 40]. Hitchcock introduces a very general *multiplex rank*, which includes tensor rank and the separate terms of our multilinear rank as special cases. A point to note is that $r_1(A)$, $r_2(A)$, $r_3(A)$, and $\operatorname{rank}_{\otimes}(A)$ are in general all different—a departure from the case of matrices, where the row rank, column rank, and outer product rank are always equal. Observe that we will always have

$$r_i(A) \le \min\{\operatorname{rank}_{\otimes}(A), d_i\}. \tag{2.16}$$

Let us verify this for r_1: if $A = \mathbf{x}_1 \otimes \mathbf{y}_1 \otimes \mathbf{z}_1 + \cdots + \mathbf{x}_r \otimes \mathbf{y}_r \otimes \mathbf{z}_r$, then each vector $A_{\bullet jk}$ belongs to $\operatorname{span}(\mathbf{x}_1, \dots, \mathbf{x}_r)$. This implies that $r_1 \le \operatorname{rank}_{\otimes}(A)$, and $r_1 \le d_1$ is immediate from the definitions. A simple but useful consequence of (2.16) is that

$$\operatorname{rank}_{\otimes}(A) \ge \|\operatorname{rank}_{\boxplus}(A)\|_\infty = \max\{r_i(A) \mid i = 1, \dots, k\}. \tag{2.17}$$

If $A \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ and $\operatorname{rank}_{\boxplus}(A) = (r_1, r_2, r_3)$, then there exist subspaces $U_i \subset \mathbb{R}^{d_i}$ with $\dim(U_i) = r_i$, such that $A \in U_1 \otimes U_2 \otimes U_3$. We call these the *supporting subspaces* of A. The supporting subspaces are minimal in the sense that if $A \in V_1 \otimes V_2 \otimes V_3$, then $U_i \subset V_i$ for $i = 1, 2, 3$. This observation leads to an alternate definition:

$$r_i(A) = \min\{\dim(U_i) \mid U_1 \subset \mathbb{R}^{d_1}, U_2 \subset \mathbb{R}^{d_2}, U_3 \subset \mathbb{R}^{d_3}, A \in U_1 \otimes U_2 \otimes U_3\}.$$

An immediate consequence of this characterization is that $\operatorname{rank}_{\boxplus}(A)$ is invariant under the action of $\mathrm{GL}_{d_1,d_2,d_3}(\mathbb{R})$: if $A' = (L, M, N) \cdot A$, where $(L, M, N) \in \mathrm{GL}_{d_1,d_2,d_3}(\mathbb{R})$, then $\operatorname{rank}_{\boxplus}(A) = \operatorname{rank}_{\boxplus}((L, M, N) \cdot A)$. Indeed, if U_1, U_2, U_3 are the supporting subspaces of A, then $L(U_1)$, $M(U_2)$, $N(U_3)$ are the supporting subspaces of $(L, M, N) \cdot A$.

More generally, we have multilinear rank equivalents of (2.14) and (2.15): if $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ and $(L_1, \dots, L_k) \in \mathbb{R}^{c_1 \times d_1} \times \cdots \times \mathbb{R}^{c_k \times d_k}$, then

$$\operatorname{rank}_{\boxplus}((L_1, \dots, L_k) \cdot A) \le \operatorname{rank}_{\boxplus}(A), \tag{2.18}$$

[2]The symbol $\boxplus$ is meant to evoke an impression of the rows and columns in a matrix.

and if $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ and $(L_1, \ldots, L_k) \in \mathrm{GL}_{d_1,\ldots,d_k}(\mathbb{R})$, then

$$\mathrm{rank}_{\boxplus}((L_1, \ldots, L_k) \cdot A) = \mathrm{rank}_{\boxplus}(A). \tag{2.19}$$

Suppose $\mathrm{rank}_{\boxplus}(A) = (r_1, r_2, r_3)$. By applying transformations $L_i \in \mathrm{GL}_{d_i}(\mathbb{R})$ which carry U_i to $\mathbb{R}^{r_i}$, it follows that A is equivalent to some $B \in \mathbb{R}^{r_1 \times r_2 \times r_3}$. Alternatively there exist $B \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ and full-rank matrices $L \in \mathbb{R}^{d_1 \times r_1}$, $M \in \mathbb{R}^{d_2 \times r_2}$, $N \in \mathbb{R}^{d_3 \times r_3}$, such that

$$A = (L, M, N) \cdot B.$$

The matrices L, M, N may be chosen to have orthonormal columns or to be unit lower-triangular—a fact easily deduced from applying the QR-decomposition or the LU-decomposition to the full-rank matrices L, M, N and using (2.8).

To a large extent, the study of tensors $A \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ with $\mathrm{rank}_{\boxplus}(A) \leq (r_1, r_2, r_3)$ reduces to the study of tensors in $\mathbb{R}^{r_1 \times r_2 \times r_3}$. This is a useful reduction, but (unlike the matrix case) it does not even come close to giving us a full classification of tensor types.

2.6. Multilinear orthogonal projection. If U is a subspace of an inner-product space V (for instance, $V = \mathbb{R}^n$ with the usual dot product), then there is an orthogonal projection from V onto U, which we denote π_U. We regard this as a map $V \to V$. As such, it is self-adjoint (i.e., has a symmetric matrix with respect to any orthonormal basis) and satisfies $\pi_U^2 = \pi_U$, $\mathrm{im}(\pi_U) = U$, $\ker(\pi_U) = U^\perp$. We note Pythagoras's theorem for any $\mathbf{v} \in V$:

$$\|\mathbf{v}\|^2 = \|\pi_U \mathbf{v}\|^2 + \|(1 - \pi_U)\mathbf{v}\|^2.$$

We now consider orthogonal projections for tensor spaces. If U_1, U_2, U_3 are subspaces of V_1, V_2, V_3, respectively, then $U_1 \otimes U_2 \otimes U_3$ is a tensor subspace of $V_1 \otimes V_2 \otimes V_3$, and the multilinear map $\Pi = (\pi_{U_1}, \pi_{U_2}, \pi_{U_3})$ is a projection onto that subspace. In fact, Π is orthogonal with respect to the Frobenius norm. The easiest way to see this is to identify $U_i \subset V_i$ with $\mathbb{R}^{c_i} \subset \mathbb{R}^{d_i}$ by taking suitable orthonormal bases; then Π acts by zeroing out all the entries of a $d_1 \times d_2 \times d_3$ array outside the initial $c_1 \times c_2 \times c_3$ block. In particular we have Pythagoras's theorem for any $A \in V_1 \otimes V_2 \otimes V_3$:

$$\|A\|_F^2 = \|\Pi A\|_F^2 + \|(1 - \Pi)A\|_F^2. \tag{2.20}$$

Being a multilinear map, Π is nonincreasing for $\mathrm{rank}_{\otimes}, \mathrm{rank}_{\boxplus}$, as in (2.14), (2.18).

There is a useful orthogonal projection Π_A associated with any tensor $A \in \mathbb{R}^{d_1 \times d_2 \times d_3}$. Let U_1, U_2, U_3 be the supporting subspaces of A so that $A \in U_1 \otimes U_2 \otimes U_3$, and $\dim(U_i) = r_i(A)$ for $i = 1, 2, 3$. Define

$$\Pi_A = (\pi_1(A), \pi_2(A), \pi_3(A)) = (\pi_{U_1}, \pi_{U_2}, \pi_{U_3}).$$

PROPOSITION 2.4. $\Pi_A(A) = A$.

Proof. A belongs to $U_1 \otimes U_2 \otimes U_3$, which is fixed by Π_A. ∎

PROPOSITION 2.5. *The function $A \mapsto \Pi_A$ is continuous over subsets of $\mathbb{R}^{d_1 \times d_2 \times d_3}$ on which $\mathrm{rank}_{\boxplus}(A)$ is constant.*

Proof. We show, for example, that $\pi_1 = \pi_1(A)$ depends continuously on A. For any $A \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, select $r = r_1(A)$ index pairs (j, k) such that the vectors $A_{\bullet jk}$ are linearly independent. For any B near A, assemble the marked vectors as a matrix

$X = X(B) \in \mathbb{R}^{d_i \times r}$. Then $\pi_1 = X(X^\top X)^{-1}X^\top =: P(B)$ by a well-known formula in linear algebra. The function $P(B)$ is defined and continuous as long as the r selected vectors remain independent, which is true on a neighborhood of A. Finally, the orthogonal projection defined by $P(B)$ maps onto the span of the r selected vectors. Thus, if $r_1(B) = r$, then $P(B) = \pi_1(B)$. ∎

It is clear that the results of this section apply to tensor spaces of all orders.

3. The algebra of tensor rank. We will state and prove a few basic results about the outer-product rank.

PROPOSITION 3.1. *Let $A \in \mathbb{R}^{c_1 \times \cdots \times c_k} \subset \mathbb{R}^{d_1 \times \cdots \times d_k}$. The rank of A regarded as a tensor in $\mathbb{R}^{c_1 \times \cdots \times c_k}$ is the same as the rank of A regarded as a tensor in $\mathbb{R}^{d_1 \times \cdots \times d_k}$.*

Proof. For each i the identity on $\mathbb{R}^{c_i}$ factors as a pair of maps $\mathbb{R}^{c_i} \overset{\iota_i}{\hookrightarrow} \mathbb{R}^{d_i} \overset{\pi_i}{\twoheadrightarrow} \mathbb{R}^{c_i}$, where ι_i is the canonical inclusion and π is the projection map given by deleting the last $d_i - c_i$ coordinates. Applying (2.14) twice, we have

$$\begin{aligned}\operatorname{rank}_\otimes(A) \geq \operatorname{rank}_\otimes((\iota_1, \dots, \iota_k) \cdot A) &\geq \operatorname{rank}_\otimes((\pi_1, \dots, \pi_k) \cdot (\iota_1, \dots, \iota_k) \cdot A) \\ &= \operatorname{rank}_\otimes((\pi_1\iota_1, \dots, \pi_k\iota_k) \cdot A) \\ &= \operatorname{rank}_\otimes(A),\end{aligned}$$

so $A \in \mathbb{R}^{c_1 \times \cdots \times c_k}$ and its image $(\iota_1, \dots, \iota_k) \cdot A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ must have equal tensor ranks. ∎

COROLLARY 3.2. *Suppose $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ and $\operatorname{rank}_\boxplus(A) \leq (c_1, \dots, c_k)$. Then $\operatorname{rank}_\otimes(A) = \operatorname{rank}_\otimes(B)$ for an equivalent tensor $B \in \mathbb{R}^{c_1 \times \cdots \times c_k}$.*

The next corollary asserts that tensor rank is consistent under a different scenario: when order-k tensors are regarded as order-l tensors for $l > k$ by taking the tensor product with a nonzero monomial term.

COROLLARY 3.3. *Let $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ be an order-k tensor and $\mathbf{u}_{k+1} \in \mathbb{R}^{d_{k+1}}, \dots, \mathbf{u}_{k+\ell} \in \mathbb{R}^{d_{k+\ell}}$ be nonzero vectors. Then*

$$\operatorname{rank}_\otimes(A) = \operatorname{rank}_\otimes(A \otimes \mathbf{u}_{k+1} \otimes \cdots \otimes \mathbf{u}_{k+\ell}).$$

Proof. Let $c_{k+1} = \cdots = c_{k+l} = 1$ and apply Proposition 3.1 to $A \in \mathbb{R}^{d_1 \times \cdots \times d_k} = \mathbb{R}^{d_1 \times \cdots \times d_k \times c_{k+1} \times \cdots \times c_{k+\ell}} \hookrightarrow \mathbb{R}^{d_1 \times \cdots \times d_k \times d_{k+1} \times \cdots \times d_{k+\ell}}$. Note that the image of the inclusion is $A \otimes \mathbf{e}_1^{(k+1)} \otimes \cdots \otimes \mathbf{e}_1^{(k+\ell)}$, where $\mathbf{e}_1^{(i)} = (1, 0, \dots, 0)^\top \in \mathbb{R}^{d_i}$. So we have

$$\operatorname{rank}_\otimes(A \otimes \mathbf{e}_1^{(k+1)} \otimes \cdots \otimes \mathbf{e}_1^{(k+\ell)}) = \operatorname{rank}_\otimes(A).$$

The general case for arbitrary nonzero $\mathbf{u}_i \in \mathbb{R}^{d_i}$ follows from applying to $A \otimes \mathbf{e}_1^{(k+1)} \otimes \cdots \otimes \mathbf{e}_1^{(k+\ell)}$ a multilinear multiplication $(I_{d_1}, \dots, I_{d_k}, L_1, \dots, L_\ell) \in \mathrm{GL}_{d_1,\dots,d_{k+\ell}}(\mathbb{R})$, where I_d is the $d \times d$ identity matrix and L_i is a nonsingular matrix with $L_i\mathbf{e}_i = \mathbf{u}_i$. It then follows from Lemma 2.3 that

$$\begin{aligned}&\operatorname{rank}_\otimes(A \otimes \mathbf{u}_{k+1} \otimes \cdots \otimes \mathbf{u}_{k+\ell}) \\ &\qquad = \operatorname{rank}_\otimes[(I_{d_1}, \dots, I_{d_k}, L_1, \dots, L_\ell) \cdot (A \otimes \mathbf{e}_1^{(k+1)} \otimes \cdots \otimes \mathbf{e}_1^{(k+\ell)})] \\ &\qquad = \operatorname{rank}_\otimes(A \otimes \mathbf{e}_1^{(k+1)} \otimes \cdots \otimes \mathbf{e}_1^{(k+\ell)}).\end{aligned}$$ ∎

Let $E = \mathbf{u}_{k+1} \otimes \mathbf{u}_{k+2} \otimes \cdots \otimes \mathbf{u}_{k+\ell} \in \mathbb{R}^{d_{k+1} \times \cdots \times d_{k+\ell}}$. So $\operatorname{rank}_\otimes(E) = 1$ and Corollary 3.3 says that $\operatorname{rank}_\otimes(A \otimes E) = \operatorname{rank}_\otimes(A) \operatorname{rank}_\otimes(E)$. Note that this last relation does not generalize. If $\operatorname{rank}_\otimes(A) > 1$ and $\operatorname{rank}_\otimes(B) > 1$, then it is true that

$$\operatorname{rank}_\otimes(A \otimes B) \leq \operatorname{rank}_\otimes(A) \operatorname{rank}_\otimes(B),$$

since one can multiply decompositions of A, B term by term to obtain a decomposition of $A \otimes B$, but it can happen (cf. [12]) that

$$\operatorname{rank}_{\otimes}(A \otimes B) < \operatorname{rank}_{\otimes}(A) \operatorname{rank}_{\otimes}(B).$$

The corresponding statement for direct sum is still an open problem for tensors of order 3 or higher. It has been conjectured by Strassen [70] that

$$\operatorname{rank}_{\otimes}(A \oplus B) \stackrel{?}{=} \operatorname{rank}_{\otimes}(A) + \operatorname{rank}_{\otimes}(B) \tag{3.1}$$

for all order-k tensors A and B. However JáJá and Takche [43] have shown that for the special case when A and B are of order 3 and at least one of them is a matrix pencil (i.e., a tensor of size $p \times q \times 2$, $p \times 2 \times q$, or $2 \times p \times q$ that may be regarded as a pair of $p \times q$ matrices), then the direct sum conjecture holds.

THEOREM 3.4 (JáJá–Takche [43]). *Let $A \in \mathbb{R}^{c_1 \times c_2 \times c_3}$ and $B \in \mathbb{R}^{d_1 \times d_2 \times d_3}$. If $2 \in \{c_1, c_2, c_3, d_1, d_2, d_3\}$, then*

$$\operatorname{rank}_{\otimes}(A \oplus B) = \operatorname{rank}_{\otimes}(A) + \operatorname{rank}_{\otimes}(B).$$

It is not hard to define tensors of arbitrarily high rank so long as we have sufficiently many linearly independent vectors in every factor.

LEMMA 3.5. *For $\ell = 1, \dots, k$, let $\mathbf{x}_1^{(\ell)}, \dots, \mathbf{x}_r^{(\ell)} \in \mathbb{R}^{d_i}$ be linearly independent. Then the tensor defined by*

$$A := \sum_{j=1}^{r} \mathbf{x}_j^{(1)} \otimes \mathbf{x}_j^{(2)} \otimes \cdots \otimes \mathbf{x}_j^{(k)} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_k}$$

has $\operatorname{rank}_{\otimes}(A) = r$.

Proof. Note that $\operatorname{rank}_{\boxplus}(A) = (r, r, \dots, r)$. By (2.17), we get

$$\operatorname{rank}_{\otimes}(A) \geq \max\{r_i(A) \mid i = 1, \dots, k\} = r.$$

On the other hand, it is clear that $\operatorname{rank}_{\otimes}(A) \leq r$. ∎

Thus, in $\mathbb{R}^{d_1 \times \cdots \times d_k}$, it is easy to write down tensors of any rank r in the range $0 \leq r \leq \min\{d_1, \dots, d_k\}$. For matrices, this exhausts all possibilities; the rank of $A \in \mathbb{R}^{d_1 \times d_2}$ is at most $\min\{d_1, d_2\}$. In contrast, for $k \geq 3$, there will always be tensors in $\mathbb{R}^{d_1 \times d_k}$ that have rank exceeding $\min\{d_1, \dots, d_k\}$. We will see this in Theorem 4.10.

4. The topology of tensor rank. Let $A = [\![a_{i_1 \cdots i_k}]\!] \in \mathbb{R}^{d_1 \times \cdots \times d_k}$. The *Frobenius norm* of A and its associated inner product are defined by

$$\|A\|_F^2 := \sum\nolimits_{i_1, \dots, i_k = 1}^{d_1, \dots, d_k} |a_{i_1 \cdots i_k}|^2, \quad \langle A, B \rangle_F := \sum\nolimits_{i_1, \dots, i_k = 1}^{d_1, \dots, d_k} a_{i_1 \cdots i_k} b_{i_1 \cdots i_k}.$$

Note that for a decomposable tensor, the Frobenius norm satisfies

$$\|\mathbf{u} \otimes \mathbf{v} \otimes \cdots \otimes \mathbf{z}\|_F = \|\mathbf{u}\|_2 \|\mathbf{v}\|_2 \cdots \|\mathbf{z}\|_2, \tag{4.1}$$

where $\|\cdot\|_2$ denotes the l^2-norm of a vector, and more generally

$$\|A \otimes B\|_F = \|A\|_F \|B\|_F \tag{4.2}$$

for arbitrary tensors A, B. Another important property which follows from (2.13) and (4.1) is orthogonal invariance:

$$\|(L_1, \dots, L_k) \cdot A\|_F = \|A\|_F \tag{4.3}$$

whenever $(L_1, \dots, L_k) \in \mathrm{O}_{d_1,\dots,d_k}(\mathbb{R})$. There are of course many other natural choices of norms on tensor product spaces [25, 36]. The important thing to note is that $\mathbb{R}^{d_1 \times \cdots \times d_k}$ being finite dimensional, all these norms will induce the same topology.

We define the following (topological) subspaces of $\mathbb{R}^{d_1 \times \cdots \times d_k}$:

$$\begin{aligned} \mathcal{S}_r(d_1, \dots, d_k) &= \left\{ A \in \mathbb{R}^{d_1 \times \cdots \times d_k} \mid \operatorname{rank}_\otimes(A) \le r \right\}, \\ \overline{\mathcal{S}}_r(d_1, \dots, d_k) &= \text{closure of } \mathcal{S}_r(d_1, \dots, d_k) \subset \mathbb{R}^{d_1 \times \cdots \times d_k}. \end{aligned}$$

Clearly the only reason to define $\overline{\mathcal{S}}_r$ is the sad fact that $\mathcal{S}_r$ is not necessarily (or even usually) closed—the theme of this paper. See section 4.2.

We occasionally refer to elements of $\mathcal{S}_r$ as "rank-r tensors." This is slightly inaccurate, since lower-rank tensors are included, but convenient. However, the direct assertions "A has rank r" and "$\operatorname{rank}(A) = r$" are always meant in the precise sense. The same remarks apply to "border rank," which is defined in section 5.5. We refer to elements of $\overline{\mathcal{S}}_r$ as "border-rank-r tensors" and describe them as being "rank-r-approximable."

Theorem 5.1 asserts that $\overline{\mathcal{S}}_2(d_1, d_2, d_3) \subset \mathcal{S}_3(d_1, d_2, d_3)$ for all d_1, d_2, d_3, and that the exceptional tensors $\overline{\mathcal{S}_2}(d_1, d_2, d_3) \setminus \mathcal{S}_2(d_1, d_2, d_3)$ are all of a particular form.

4.1. Upper semicontinuity. Discrete-valued rank functions on spaces of matrices or tensors cannot usefully be continuous, because they would then be constant and would not have any classifying power. As a sort of compromise, matrix rank is well known to be an upper semicontinuous function; if $\operatorname{rank}(A) = r$, then $\operatorname{rank}(B) \ge r$ for all matrices B in a neighborhood of A. This is not true for the outer-product rank of tensors (as we will see in section 4.2). There are several equivalent ways of formulating this assertion.

PROPOSITION 4.1. *Let $r \ge 2$ and $k \ge 3$. Given the norm-topology on $\mathbb{R}^{d_1 \times \cdots \times d_k}$, the following statements are equivalent:*

(a) *The set $\mathcal{S}_r(d_1, \dots, d_k) := \{A \in \mathbb{R}^{d_1 \times \cdots \times d_k} \mid \operatorname{rank}_\otimes(A) \le r\}$ is not closed.*

(b) *There exists a sequence $A_n \in \mathbb{R}^{d_1 \times \cdots \times d_k}$, $\operatorname{rank}_\otimes(A_n) \le r$, $n \in \mathbb{N}$, converging to $B \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ with $\operatorname{rank}_\otimes(B) > r$.*

(c) *There exists $B \in \mathbb{R}^{d_1 \times \cdots \times d_k}$, $\operatorname{rank}_\otimes(B) > r$, that may be approximated arbitrarily closely by tensors of strictly lower rank, i.e.,*

$$\inf\{\|B - A\| \mid \operatorname{rank}_\otimes(A) \le r\} = 0.$$

(d) *There exists $C \in \mathbb{R}^{d_1 \times \cdots \times d_k}$, $\operatorname{rank}_\otimes(C) > r$, that does not have a best rank-r approximation; i.e.,*

$$\inf\{\|C - A\| \mid \operatorname{rank}_\otimes(A) \le r\}$$

is not attained (by any A with $\operatorname{rank}_\otimes(A) \le r$).

Proof. It is obvious that (a) $\Rightarrow$ (b) $\Rightarrow$ (c) $\Rightarrow$ (d). To complete the chain, we just need to show that (d) $\Rightarrow$ (a). Suppose $\mathcal{S} := \mathcal{S}_r(d_1, \dots, d_k)$ is closed. Since the closed ball of radius $\|C\|$ centered at C, $\{A \in \mathbb{R}^{d_1 \times \cdots \times d_k} \mid \|C - A\| \le \|C\|\}$, intersects

$\mathcal{S}$ nontrivially (e.g., 0 is in both sets), their intersection $\mathcal{T} = \{A \in \mathbb{R}^{d_1 \times \cdots \times d_k} \mid \operatorname{rank}_\otimes(A) \leq r, \|C - A\| \leq \|C\|\}$ is a nonempty compact set. Now observe that

$$\delta := \inf\{\|C - A\| \mid A \in \mathcal{S}\} = \inf\{\|C - A\| \mid A \in \mathcal{T}\}$$

since any $A' \in \mathcal{S} \backslash \mathcal{T}$ must have $\|C - A'\| > \|C\|$ while we know that $\delta \leq \|C\|$. By the compactness of $\mathcal{T}$, there exists $A_* \in \mathcal{T}$ such that $\|C - A_*\| = \delta$. So the required infimum is attained by $A_* \in \mathcal{T} \subset \mathcal{S}$. ∎

We caution the reader that there exist tensors of rank $> r$ that do not have a best rank-r approximation but *cannot* be approximated arbitrarily closely by rank-r tensors, i.e., $\inf\{\|C - A\| \mid \operatorname{rank}_\otimes(A) \leq r\} > 0$. In other words, statement (d) applies to a strictly larger class of tensors than statement (c) (cf. section 8). The tensors in statement (d) are sometimes called "degenerate" in the psychometrics and chemometrics literature (e.g., [49, 51, 63, 68, 69]), but we prefer to avoid this term since it is inconsistent (and often at odds) with common usage in Mathematics. For example, in Table 7.1, the tensors in the orbit classes of D_2, D_2', D_2'' are all degenerate, but statement (d) does not apply to them; on the other hand, the tensors in the orbit class of G_3 are nondegenerate, but Theorem 8.1 tells us that they are all of the form in statement (d).

We begin by getting three well-behaved cases out of the way. The proofs shed light on what can go wrong in all the other cases.

PROPOSITION 4.2. *For all $d_1, \ldots, d_k$, we have $\overline{\mathcal{S}}_1(d_1, \ldots, d_k) = \mathcal{S}_1(d_1, \ldots, d_k)$.*

Proof. Suppose $A_n \to A$, where $\operatorname{rank}_\otimes(A_n) \leq 1$. We can write

$$A_n = \lambda_n \mathbf{u}_{1,n} \otimes \mathbf{u}_{2,n} \otimes \cdots \otimes \mathbf{u}_{k,n},$$

where $\lambda_n = \|A_n\|$ and the vectors $\mathbf{u}_{i,n} \in \mathbb{R}^{d_i}$ have unit norm. Certainly $\lambda_n = \|A_n\| \to \|A\| =: \lambda$. Moreover, since the unit sphere in $\mathbb{R}^{d_i}$ is compact, each sequence $\mathbf{u}_{i,n}$ has a convergent subsequence, with limit $\mathbf{u}_i$, say. It follows that there is a subsequence of A_n which converges to $\lambda \mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_k$. This must equal A, and it has rank at most 1. ∎

PROPOSITION 4.3. *For all r and d_1, d_2, we have $\overline{\mathcal{S}}_r(d_1, d_2) = \mathcal{S}_r(d_1, d_2)$. In other words, matrix rank is upper-semicontinuous.*

Proof. Suppose $A_n \to A$, where $\operatorname{rank}(A_n) \leq r$, so we can write

$$A_n = \lambda_{1,n} \mathbf{u}_{1,n} \otimes \mathbf{v}_{1,n} + \cdots + \lambda_{r,n} \mathbf{u}_{r,n} \otimes \mathbf{v}_{r,n}.$$

Convergence of the sequence A_n does not imply convergence of the individual terms $\lambda_{i,n}$, $\mathbf{u}_{i,n}$, $\mathbf{v}_{i,n}$, even in a subsequence. However, if we take the singular value decomposition, then the $\mathbf{u}_{i,n}$ and $\mathbf{v}_{i,n}$ are unit vectors and the $\lambda_{i,n}$ satisfy

$$\lambda_{1,n}^2 + \cdots + \lambda_{r,n}^2 = \|A_n\|.$$

Since $\|A_n\| \to \|A\|$ this implies that the $\lambda_{i,n}$ are uniformly bounded. Thus we can find a subsequence with convergence $\lambda_{i,n} \to \lambda_i$, $\mathbf{u}_{i,n} \to \mathbf{u}_i$, $\mathbf{v}_{i,n} \to \mathbf{v}_i$ for all i. Then

$$A = \lambda_1 \mathbf{u}_1 \otimes \mathbf{v}_1 + \cdots + \lambda_r \mathbf{u}_r \otimes \mathbf{v}_r,$$

which has rank at most r. ∎

PROPOSITION 4.4. *The multilinear rank function $\operatorname{rank}_\boxplus(A) = (r_1(A), \ldots, r_k(A))$ is upper-semicontinuous.*

Proof. Each r_i is the rank of a matrix obtained by rearranging the entries of A, and is therefore upper-semicontinuous in A by Proposition 4.3. ∎

COROLLARY 4.5. *Every tensor has a best rank-1 approximation. Every matrix has a best rank-r approximation. Every order-k tensor has a best approximation with* $\operatorname{rank}_{\boxplus} \leq (r_1, \dots, r_k)$ *for any specified* $(r_1, \dots, r_k)$.

Proof. These statements follow from Propositions 4.2, 4.3, and 4.4, together with the implication (d)⇒(a) from Proposition 4.1. ∎

4.2. Tensor rank is not upper-semicontinuous. Here is the simplest example of the failure of outer-product rank to be upper-semicontinuous. This is the first example of a more general construction which we discuss in section 4.7. A formula similar to (4.4) appeared as Exercise 62 in section 4.6.4 of Knuth's *The Art of Computer Programming* [48] (the original source is [8]). Other examples have appeared in [7] (the earliest known to us) and [63], as well as in unpublished work of Kruskal.

PROPOSITION 4.6. *Let* $\mathbf{x}_1, \mathbf{y}_1 \in \mathbb{R}^{d_1}$, $\mathbf{x}_2, \mathbf{y}_2 \in \mathbb{R}^{d_2}$, *and* $\mathbf{x}_3, \mathbf{y}_3 \in \mathbb{R}^{d_3}$ *be vectors such that each pair* $\mathbf{x}_i, \mathbf{y}_i$ *is linearly independent. Then the tensor*

$$A := \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{y}_3 + \mathbf{x}_1 \otimes \mathbf{y}_2 \otimes \mathbf{x}_3 + \mathbf{y}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \in \mathbb{R}^{d_1 \times d_2 \times d_3} \tag{4.4}$$

has rank 3 *but can be approximated arbitrarily closely by tensors of rank* 2. *In particular, A does not have a best rank-*2 *approximation.*

Proof. For each $n \in \mathbb{N}$, define

$$A_n := n\left(\mathbf{x}_1 + \frac{1}{n}\mathbf{y}_1\right) \otimes \left(\mathbf{x}_2 + \frac{1}{n}\mathbf{y}_2\right) \otimes \left(\mathbf{x}_3 + \frac{1}{n}\mathbf{y}_3\right) - n\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3. \tag{4.5}$$

Clearly, $\operatorname{rank}_{\otimes}(A_n) \leq 2$, and since, as $n \to \infty$,

$$\|A_n - A\|_F \leq \frac{1}{n}\|\mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{x}_3 + \mathbf{y}_1 \otimes \mathbf{x}_2 \otimes \mathbf{y}_3 + \mathbf{x}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3\|_F + \frac{1}{n^2}\|\mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3\|_F \to 0,$$

we see that A is approximated arbitrary closely by tensors A_n.

It remains to establish that $\operatorname{rank}_{\otimes}(A) = 3$. From the three-term format of A, we deduce only that $\operatorname{rank}_{\otimes}(A) \leq 3$. A clean proof that $\operatorname{rank}_{\otimes}(A) > 2$ is included in the proof of Theorem 7.1, but this depends on the properties of the polynomial Δ defined in section 5.3. A more direct argument is given in the next lemma. ∎

LEMMA 4.7. *Let* $\mathbf{x}_1, \mathbf{y}_1 \in \mathbb{R}^{d_1}$, $\mathbf{x}_2, \mathbf{y}_2 \in \mathbb{R}^{d_2}$, $\mathbf{x}_3, \mathbf{y}_3 \in \mathbb{R}^{d_3}$, *and*

$$A = \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{y}_3 + \mathbf{x}_1 \otimes \mathbf{y}_2 \otimes \mathbf{x}_3 + \mathbf{y}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3.$$

Then $\operatorname{rank}_{\otimes}(A) = 3$ *if and only if* $\mathbf{x}_i, \mathbf{y}_i$ *are linearly independent for* $i = 1, 2, 3$.

Proof. Only two distinct vectors are involved in each factor of the tensor product, so $\operatorname{rank}_{\boxplus}(A) \leq (2, 2, 2)$ and we can work in $\mathbb{R}^{2\times 2\times 2}$ (Corollary 3.2). More strongly, if any of the pairs $\{\mathbf{x}_i, \mathbf{y}_i\}$ is linearly dependent, then A is GL-equivalent to a tensor in $\mathbb{R}^{1\times 2\times 2}$, $\mathbb{R}^{2\times 1\times 2}$, or $\mathbb{R}^{2\times 2\times 1}$. These spaces are isomorphic to $\mathbb{R}^{2\times 2}$, so the maximum possible rank of A is 2.

Conversely, suppose each pair $\{\mathbf{x}_i, \mathbf{y}_i\}$ is linearly independent. We may as well assume that

$$A = \left[\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{array}\right] \tag{4.6}$$

since we can transform A to that form using a multilinear transformation (L_1, L_2, L_3), where $L_i(\mathbf{x}_i) = \mathbf{e}_1$ and $L_i(\mathbf{y}_i) = \mathbf{e}_2$ for $i = 1, 2, 3$.

Suppose, for a contradiction, that $\operatorname{rank}_\otimes(A) \leq 2$; then we can write

$$A = \mathbf{u}_1 \otimes \mathbf{u}_2 \otimes \mathbf{u}_3 + \mathbf{v}_1 \otimes \mathbf{v}_2 \otimes \mathbf{v}_3 \tag{4.7}$$

for some $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{R}^{d_i}$.

Claim 1. The vectors $\mathbf{u}_1, \mathbf{v}_1$ are independent. If they are not, then let $\varphi : \mathbb{R}^2 \to \mathbb{R}$ be a nonzero linear map such that $\varphi(\mathbf{u}_1) = \varphi(\mathbf{v}_1) = 0$. Using the expressions in (4.7) and (4.6), we find that

$$\mathbf{0} = (\varphi, I, I) \cdot A = \begin{bmatrix} \varphi(\mathbf{e}_2) & \varphi(\mathbf{e}_1) \\ \varphi(\mathbf{e}_1) & 0 \end{bmatrix}$$

in $\mathbb{R}^{1\times 2\times 2} \cong \mathbb{R}^{2\times 2}$, which is a contradiction because $\varphi(\mathbf{e}_1)$ and $\varphi(\mathbf{e}_2)$ cannot both be zero.

Claim 2. The vectors $\mathbf{u}_1, \mathbf{e}_1$ are dependent. Indeed, let $\varphi_u : \mathbb{R}^2 \to \mathbb{R}$ be a linear map whose kernel is spanned by $\mathbf{u}_1$. Then

$$\varphi_u(\mathbf{v}_1)(\mathbf{v}_2 \otimes \mathbf{v}_3) = (\varphi_u, I, I) \cdot A = \begin{bmatrix} \varphi_u(\mathbf{e}_2) & \varphi_u(\mathbf{e}_1) \\ \varphi_u(\mathbf{e}_1) & 0 \end{bmatrix}$$

in $\mathbb{R}^{1\times 2\times 2} \cong \mathbb{R}^{2\times 2}$. The left-hand side (LHS) has rank at most 1, which implies on the right-hand side (RHS) that $\varphi_u(\mathbf{e}_1) = 0$, and hence $\mathbf{e}_1 \in \operatorname{span}\{\mathbf{u}_1\}$.

Claim 3. The vectors $\mathbf{v}_1, \mathbf{e}_1$ are dependent. Indeed, let $\varphi_v : \mathbb{R}^2 \to \mathbb{R}$ be a linear map whose kernel is spanned by $\mathbf{v}_1$. Then

$$\varphi_v(\mathbf{u}_1)(\mathbf{u}_2 \otimes \mathbf{u}_3) = (\varphi_v, I, I) \cdot A = \begin{bmatrix} \varphi_v(\mathbf{e}_2) & \varphi_v(\mathbf{e}_1) \\ \varphi_v(\mathbf{e}_1) & 0 \end{bmatrix}$$

in $\mathbb{R}^{1\times 2\times 2} \cong \mathbb{R}^{2\times 2}$. The LHS has rank at most 1, which implies on the RHS that $\varphi_v(\mathbf{e}_1) = 0$, and hence $\mathbf{e}_1 \in \operatorname{span}\{\mathbf{v}_1\}$.

Taken together, the three claims are inconsistent. This is the desired contradiction. Thus $\operatorname{rank}_\otimes(A) > 2$, and therefore $\operatorname{rank}_\otimes(A) = 3$. ∎

Remark. Note that if we take $d_1 = d_2 = d_3 = 2$, then (4.4) is an example of a tensor whose outer-product rank exceeds $\min\{d_1, d_2, d_3\}$.

4.3. Diverging coefficients. What goes wrong in the example of Proposition 4.6? Why do the rank-2 decompositions of the A_n fail to converge to a rank-2 decomposition of A? We can attempt to mimic the proofs of Propositions 4.2 and 4.3 by seeking convergent subsequences for the rank-2 decompositions of the A_n. We fail because we cannot simultaneously keep all the variables bounded. For example, in the decomposition

$$A_n = n\left(\mathbf{x}_1 + \frac{1}{n}\mathbf{y}_1\right) \otimes \left(\mathbf{x}_2 + \frac{1}{n}\mathbf{y}_2\right) \otimes \left(\mathbf{x}_3 + \frac{1}{n}\mathbf{y}_3\right) - n\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3$$

the vector terms converge but the coefficients $\lambda_1 = \lambda_2 = n$ tend to infinity. In spite of this, the sequence A_n itself remains bounded.

In fact, rank-jumping always occurs like this (see also [49]).

PROPOSITION 4.8. *Suppose* $A_n \to A$, *where* $\operatorname{rank}_\otimes(A) \geq r+1$ *and* $\operatorname{rank}_\otimes(A_n) \leq r$ *for all* n. *If we write*

$$A_n = \lambda_{1,n}\mathbf{u}_{1,n} \otimes \mathbf{v}_{1,n} \otimes \mathbf{w}_{1,n} + \cdots + \lambda_{r,n}\mathbf{u}_{r,n} \otimes \mathbf{v}_{r,n} \otimes \mathbf{w}_{r,n},$$

where the vectors $\mathbf{u}_{i,n}$, $\mathbf{v}_{i,n}$, $\mathbf{w}_{i,n}$ *are unit vectors, then* $\max_i\{|\lambda_{i,n}|\} \to \infty$ *as* $n \to \infty$. *Moreover, at least two of the coefficient sequences* $\{\lambda_{i,n} \mid n = 1, 2, \dots\}$ *are unbounded.*

Proof. If the sequence $\max_i\{|\lambda_{i,n}|\}$ does not diverge to ∞, then it has a bounded subsequence. In this subsequence, the coefficients and vectors are all bounded, so we can pass to a further subsequence in which each of the coefficient sequences and vector sequences is convergent:

$$\lambda_{i,n} \to \lambda_i, \quad \mathbf{u}_{i,n} \to \mathbf{u}_i, \quad \mathbf{v}_{i,n} \to \mathbf{v}_i, \quad \mathbf{w}_{i,n} \to \mathbf{w}_i.$$

It follows that $A = \lambda_1\mathbf{u}_1 \otimes \mathbf{v}_1 \otimes \mathbf{w}_1 + \cdots + \lambda_r\mathbf{u}_r \otimes \mathbf{v}_r \otimes \mathbf{w}_r$, so it has rank at most r, which is a contradiction.

Thus $\max_i\{|\lambda_{i,n}|\}$ diverges to ∞. It follows that at least one of the coefficient sequences has a divergent subsequence. If there were only one such coefficient sequence, all the others being bounded, then (on the subsequence) A_n would be dominated by this term and consequently $\|A_n\|$ would be unbounded. Since $A_n \to A$, this cannot happen. Thus there are at least two unbounded coefficient sequences. ∎

For a *minimal* rank-jumping example, *all* the coefficients must diverge to ∞.

PROPOSITION 4.9. *Suppose* $A_n \to A$, *where* $\operatorname{rank}_\otimes(A) = r+s$ *and* $\operatorname{rank}_\otimes(A_n) \le r$ *for all* n. *If we write*

$$A_n = \lambda_{1,n}\mathbf{u}_{1,n} \otimes \mathbf{v}_{1,n} \otimes \mathbf{w}_{1,n} + \cdots + \lambda_{r,n}\mathbf{u}_{r,n} \otimes \mathbf{v}_{r,n} \otimes \mathbf{w}_{r,n},$$

where the vectors $\mathbf{u}_{i,n}$, $\mathbf{v}_{i,n}$, $\mathbf{w}_{i,n}$ *are unit vectors, then there are two possibilities: either* (i) *all of the sequences* $|\lambda_{i,n}|$ *diverge to* ∞ *as* $n \to \infty$ *or* (ii) *in the same tensor space there exists* $B_n \to B$, *where* $\operatorname{rank}_\otimes(B) \ge r' + s$ *and* $\operatorname{rank}_\otimes(B_n) \le r'$ *for all* n, *for some* $r' < r$.

Proof. Suppose one of the coefficient sequences, say, $|\lambda_{i,n}|$, fails to diverge as $n \to \infty$; so it has a bounded subsequence. In a further subsequence, the ith term $R_n = \lambda_{i,n}\mathbf{u}_{i,n} \otimes \mathbf{v}_{i,n} \otimes \mathbf{w}_{i,n}$ converges to a tensor R of rank (at most) 1. Writing $B_n = A_n - R_n$, we find that $B_n \to B = A - R$ on this subsequence, with $\operatorname{rank}_\otimes(B_n) \le r-1$. Moreover, $r+s \le \operatorname{rank}_\otimes(A) \le \operatorname{rank}_\otimes(B) + \operatorname{rank}_\otimes(R)$, so $\operatorname{rank}_\otimes(B) \ge (r-1)+s$. ∎

Remark. Clearly the arguments in Propositions 4.8 and 4.9 apply to tensors of all orders, not just order 3. We also note that the vectors ($\mathbf{u}_{i,n}$, etc.) need not be unit vectors; they just have to be uniformly bounded.

One interpretation of Proposition 4.8 is that if one attempts to minimize

$$\|A - \lambda_1\mathbf{u}_1 \otimes \mathbf{v}_1 \otimes \mathbf{w}_1 - \cdots - \lambda_r\mathbf{u}_r \otimes \mathbf{v}_r \otimes \mathbf{w}_r\|$$

for a tensor A which does not have a best rank-r approximation, then (at least some of) the coefficients λ_i become unbounded. This phenomenon of diverging summands has been observed in practical applications of multilinear models in psychometrics and chemometrics and is commonly referred to in those circles as "CANDECOMP/PARAFAC degeneracy" or "diverging CANDECOMP/PARAFAC components" [49, 51, 63, 68, 69]. More precisely, these are called "k-factor degeneracies" when there are k diverging summands whose sum stays bounded. 2- and 3-factor degeneracies were exhibited in [63] and 4- and 5-factor degeneracies were exhibited in [68]. There are uninteresting (see section 4.4) and interesting (see section 4.7) ways of generating k-factor degeneracies for arbitrarily large k.

4.4. Higher orders, higher ranks, arbitrary norms. We will now show that the rank-jumping phenomenon—that is, the failure of $\mathcal{S}_r(d_1, \dots, d_k)$ to be closed—is independent of the choice of norms and can be extended to arbitrary order. The norm

independence is a trivial consequence of a basic fact in functional analysis: all norms on finite dimensional vector spaces are equivalent; in particular, any norm will induce the same unique topology on a finite dimensional vector space.

THEOREM 4.10. *For $k \geq 3$ and $d_1, \dots, d_k \geq 2$, the problem of determining a best rank-r approximation for an order-k tensor in $\mathbb{R}^{d_1 \times \cdots \times d_k}$ has no solution in general for any $r = 2, \dots, \min\{d_1, \dots, d_k\}$. In particular, there exists $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ with*

$$\operatorname{rank}_{\otimes}(A) = r + 1$$

that has no best rank-r approximation. The result is independent of the choice of norms.

Proof. We begin by assuming $k = 3$.

Higher rank. Let $2 \leq r \leq \min\{d_1, d_2, d_3\}$. By Lemma 3.5, we can construct a tensor $B \in \mathbb{R}^{(d_1-2)\times(d_2-2)\times(d_3-2)}$ with rank $r-2$. By Proposition 4.6, we can construct a convergent sequence of tensors $C_n \to C$ in $\mathbb{R}^{2\times 2\times 2}$ with $\operatorname{rank}_{\otimes}(C_n) \leq 2$ and $\operatorname{rank}_{\otimes}(C) = 3$. Let $A_n = B \oplus C_n \in \mathbb{R}^{d_1 \times d_2 \times d_3}$. Then $A_n \to A := B \oplus C$ and $\operatorname{rank}_{\otimes}(A_n) \leq \operatorname{rank}_{\otimes}(B) + \operatorname{rank}_{\otimes}(C_n) \leq r$. The result of JáJá–Takche (Theorem 3.4) implies that $\operatorname{rank}_{\otimes}(A) = \operatorname{rank}_{\otimes}(B) + \operatorname{rank}_{\otimes}(C) = r + 1$.

Arbitrary order. Let $\mathbf{u}_4 \in \mathbb{R}^{d_4}, \dots, \mathbf{u}_k \in \mathbb{R}^{d_k}$ be unit vectors and set

$$\tilde{A}_n := A_n \otimes \mathbf{u}_4 \otimes \cdots \otimes \mathbf{u}_k, \qquad \tilde{A} := A \otimes \mathbf{u}_4 \otimes \cdots \otimes \mathbf{u}_k.$$

By (4.2),

$$\|\tilde{A}_n - \tilde{A}\|_F = \|A_n - A\| = \|B \oplus C_n - B \oplus C\| = \|C_n - C\| \to 0 \text{ as } n \to \infty.$$

Moreover, Corollary 3.3 ensures that $\operatorname{rank}_{\otimes}(\tilde{A}) = r + 1$ and $\operatorname{rank}_{\otimes}(\tilde{A}_n) \leq r$.

Norm independence. Whether the sequence $\tilde{A}_n$ converges to $\tilde{A}$ is entirely dependent on the norm-induced topology on $\mathbb{R}^{d_1 \times \cdots \times d_k}$. Since it has a unique topology induced by any of its equivalent norms as a finite dimensional vector space, the convergence is independent of the choice of norms. ∎

We note that the proof above exhibits an order-k tensor, namely, $\tilde{A}$, that has rank strictly larger than $\min\{d_1, \dots, d_k\}$.

4.5. Tensor rank can leap an arbitrarily large gap. How can we construct a sequence of tensors of rank r that converge to a tensor of rank $r+2$? An easy trick is to take the direct sum of two sequences of rank-2 tensors of the form shown in (4.5). The resulting sequence converges to a limiting tensor that is the direct sum of two rank-3 tensors, each of the form shown in (4.4). To show that the limiting tensor has rank 6 (and does not have some miraculous lower-rank decomposition), we once again turn to the theorem of JáJá–Takche, which contains just enough of the direct sum conjecture (3.1) for our purposes.

PROPOSITION 4.11. *Given any $s \in \mathbb{N}$ and $r \geq 2s$, there exists a sequence of order-3 tensors B_n such that $\operatorname{rank}_{\otimes}(B_n) \leq r$ and $\lim_{n\to\infty} B_n = B$ with $\operatorname{rank}_{\otimes}(B) = r + s$.*

Proof. Let $d = r - 2s$. By Lemma 3.5, there exists a rank-d tensor $C \in \mathbb{R}^{d\times d\times d}$. Let $A_n \to A$ be a convergent sequence in $\mathbb{R}^{2\times 2\times 2}$ with $\operatorname{rank}_{\otimes}(A) \leq 2$ and $\operatorname{rank}_{\otimes}(A) = 3$. Define

$$B_n = C \oplus A_n \oplus \cdots \oplus A_n, \quad B = C \oplus A \oplus \cdots \oplus A,$$

where there are s terms A_n and A. Then $B_n \to B$, and $\operatorname{rank}_{\otimes}(B_n) \leq r - 2s + 2s = r$. By applying the JáJá–Takche theorem sequentially s times, once for each summand A, we deduce that $\operatorname{rank}_{\otimes}(B) = r - 2s + 3s = r + s$. ∎

As usual the construction can be extended to order-k tensors by taking an outer product with a suitable number of nonzero vectors in the new factors.

COROLLARY 4.12. *Given any $s \geq 1$, $r \geq 2$, and $k \geq 3$, with $r \geq 2s$, there exists $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ such that $\operatorname{rank}_{\otimes}(A) = r + s$ and A has no best rank-r approximation.*

Proof. This follows from Proposition 4.11 and the previous remark. ∎

4.6. Brègman divergences and other continuous measures of proximity. In data analytic applications, one frequently encounters low-rank approximations with respect to "distances" that are more general than norms. Such a "distance" may not even be a metric, an example being the Brègman divergence [10, 26] (sometimes also known as the Brègman distance). The definition here is based on the definition given in [26]. Recall first that if $S \subset \mathbb{R}^n$, the *relative interior* of S is simply the interior of S considered as a subset of its affine hull and is denoted by $\operatorname{ri}(S)$.

DEFINITION 4.13. *Let $S \subseteq \mathbb{R}^{d_1 \times \cdots \times d_k}$ be a convex set. Let $\varphi : S \to \mathbb{R}$ be a lower-semicontinuous, convex function that is continuously differentiable and strictly convex in $\operatorname{ri}(S)$. Let φ have the property that for any sequence $\{C_n\} \subset \operatorname{ri}(S)$ that converges to $C \in S \setminus \operatorname{ri}(S)$, we have*

$$\lim_{n\to\infty} \|\nabla\varphi(C_n)\| = +\infty.$$

The Brègman divergence $D_\varphi : S \times \operatorname{ri}(S) \to \mathbb{R}$ *is defined by*

$$D_\varphi(A, B) = \varphi(A) - \varphi(B) - \langle \nabla\varphi(B), A - B\rangle.$$

It is natural to ask if the analogous problem APPROX(A, r) for Brègman divergence will always have a solution. Note that a Brègman divergence, unlike a metric, is not necessarily symmetric in its two arguments, and thus there are *two* possible problems:

$$\operatorname{argmin}_{\operatorname{rank}_{\otimes}(B)\leq r} D_\varphi(A, B) \qquad \text{and} \qquad \operatorname{argmin}_{\operatorname{rank}_{\otimes}(B)\leq r} D_\varphi(B, A).$$

As the following proposition shows, the answer is no in both cases.

PROPOSITION 4.14. *Let D_φ be a Brègman divergence. Let A and A_n be defined as in* (4.4) *and* (4.5)*, respectively. Then*

$$\lim_{n\to\infty} D_\varphi(A, A_n) = 0 = \lim_{n\to\infty} D_\varphi(A_n, A).$$

Proof. The Brègman divergence is jointly continuous in both arguments with respect to the norm topology, and $A_n \to A$ in the norm, so $D_\varphi(A, A_n) \to D_\varphi(A, A) = 0$ and $D_\varphi(A_n, A) \to D_\varphi(A, A) = 0$. ∎

Proposition 4.14 extends trivially to any other measure of nearness that is continuous with respect to the norm topology in at least one argument.

4.7. Difference quotients. We thank Landsberg [53] for the insight that the expression in (4.4) is best regarded as a derivative. Indeed, if

$$f(t) = (\mathbf{x} + t\mathbf{y})^{\otimes 3} = (\mathbf{x} + t\mathbf{y}) \otimes (\mathbf{x} + t\mathbf{y}) \otimes (\mathbf{x} + t\mathbf{y}),$$

then

$$\left.\frac{df}{dt}\right|_{t=0} = \mathbf{y} \otimes \mathbf{x} \otimes \mathbf{x} + \mathbf{x} \otimes \mathbf{y} \otimes \mathbf{x} + \mathbf{x} \otimes \mathbf{x} \otimes \mathbf{y}$$

by the Leibniz rule. On the other hand,

$$\left.\frac{df}{dt}\right|_{t=0} = \lim_{t\to 0}\left[\frac{(\mathbf{x}+t\mathbf{y})\otimes(\mathbf{x}+t\mathbf{y})\otimes(\mathbf{x}+t\mathbf{y}) - \mathbf{x}\otimes\mathbf{x}\otimes\mathbf{x}}{t}\right],$$

and the difference quotient on the RHS has rank 2. The expression in (4.5) can be obtained from this by taking $t = 1/N$.

We can extend Landsberg's idea to more general partial differential operators. It will be helpful to use the degree-k Veronese map [37], which is $V_k(\mathbf{x}) = \mathbf{x}^{\otimes k} = \mathbf{x}\otimes\cdots\otimes\mathbf{x}$ (a k-fold product). Then, for example, the six-term symmetric tensor

$$\mathbf{x}\otimes\mathbf{y}\otimes\mathbf{z} + \mathbf{x}\otimes\mathbf{z}\otimes\mathbf{y} + \mathbf{y}\otimes\mathbf{z}\otimes\mathbf{x} + \mathbf{y}\otimes\mathbf{x}\otimes\mathbf{z} + \mathbf{z}\otimes\mathbf{x}\otimes\mathbf{y} + \mathbf{x}\otimes\mathbf{y}\otimes\mathbf{x}$$

can be written as a partial derivative

$$\left.\frac{\partial^2}{\partial s\,\partial t}\right|_{s=t=0} (\mathbf{x}+s\mathbf{y}+t\mathbf{z})^{\otimes 3},$$

which is a limit of a four-term difference quotient:

$$\lim_{s,t\to 0}\left[\frac{V_3(\mathbf{x}+s\mathbf{y}+t\mathbf{z}) - V_3(\mathbf{x}+s\mathbf{y}) - V_3(\mathbf{x}+t\mathbf{z}) + V_3(\mathbf{x})}{st}\right].$$

This example lies naturally in $\mathbb{R}^{3\times 3\times 3}$, taking $\mathbf{x},\mathbf{y},\mathbf{z}$ to be linearly independent. Another example, in $\mathbb{R}^{2\times 2\times 2\times 2}$, is the six-term symmetric order-4 tensor

$$\begin{aligned}\mathbf{x}\otimes\mathbf{x}\otimes\mathbf{y}\otimes\mathbf{y} + \mathbf{x}\otimes\mathbf{y}\otimes\mathbf{x}\otimes\mathbf{y} + \mathbf{x}\otimes\mathbf{y}\otimes\mathbf{y}\otimes\mathbf{x}\\ + \mathbf{y}\otimes\mathbf{x}\otimes\mathbf{x}\otimes\mathbf{y} + \mathbf{y}\otimes\mathbf{x}\otimes\mathbf{y}\otimes\mathbf{x} + \mathbf{y}\otimes\mathbf{y}\otimes\mathbf{x}\otimes\mathbf{x}.\end{aligned}$$

This can be written as the second-order derivative

$$\left.\frac{\partial^2}{\partial t^2}\right|_{t=0} \frac{(\mathbf{x}+t\mathbf{y})^{\otimes 4}}{2!},$$

which is a limit of a three-term difference quotient:

$$\lim_{t\to 0}\left[\frac{V_4(\mathbf{x}+2t\mathbf{y}) - 2V_4(\mathbf{x}+t\mathbf{y}) + V_4(\mathbf{x})}{2!\,t^2}\right].$$

We call these examples *symmetric Leibniz tensors* for the differential operators $\partial^2/\partial s\,\partial t$ and $\partial^2/\partial t^2$, of orders 3 and 4, respectively. More generally, given positive integers k and $a_1,\dots,a_j$ with $a_1+\cdots+a_j = a \le k$, the symmetric tensor

$$L_k(a_1,\dots,a_j) := \sum_{\text{Sym}} \mathbf{x}^{\otimes(k-a)}\otimes\mathbf{y}_1^{\otimes a_1}\otimes\cdots\otimes\mathbf{y}_j^{\otimes a_j}$$

can be written as a partial derivative,

$$\left.\frac{\partial^a}{\partial {t_1}^{a_1}\dots\partial {t_j}^{a_j}}\right|_{t_1=\dots=t_j=0} \frac{V_k(\mathbf{x}+t_1\mathbf{y}_1+\cdots+t_j\mathbf{y}_j)}{(a_1!)\cdots(a_j!)},$$

which is a limit of a difference quotient with $(a_1+1)\cdots(a_j+1)$ terms. On the other hand, the number of terms in the limit $L_k(a_1,\dots,a_j)$ is given by a multinomial coefficient, and that is usually much bigger.

This construction gives us a ready supply of candidates for rank-jumping. However, we do not know—even for the two explicit six-term examples above — whether the limiting tensors actually have the ranks suggested by their formulas. We can show that $\operatorname{rank}_\otimes(L_k(1)) = k$ for all k and over any field, generalizing Lemma 4.7. Beyond that it is not clear to us what is likely to be true. The optimistic conjecture is

$$\operatorname{rank}_\otimes(L_k(a_1,\dots,a_j)) \stackrel{?}{=} \binom{k}{k-a,a_1,\dots,a_j} = \frac{k!}{(k-a)!\,a_i!\cdots a_j!}. \tag{4.8}$$

Comon et al. [18] show that the *symmetric* rank of $L_k(1)$ over the complex numbers is k, so that is another possible context in which (4.8) may be true.

5. Characterizing the limit points of order-3 rank-2 tensors. If an order-3 tensor can be expressed as a limit of a sequence of rank-2 tensors but itself has rank greater than 2, then we show in this section that it takes a particular form. This kind of result may make it possible to overcome the ill-posedness of $\text{APPROX}(A, r)$ by defining weak solutions.

THEOREM 5.1. *Let $d_1, d_2, d_3 \geq 2$. Let $A_n \in \mathbb{R}^{d_1\times d_2\times d_3}$ be a sequence of tensors with $\operatorname{rank}_\otimes(A_n) \leq 2$ and*

$$\lim_{n\to\infty} A_n = A,$$

where the limit is taken in any norm topology. If the limiting tensor A has rank higher than 2, then $\operatorname{rank}_\otimes(A)$ must be exactly 3, and there exist pairs of linearly independent vectors $\mathbf{x}_1, \mathbf{y}_1 \in \mathbb{R}^{d_1}$, $\mathbf{x}_2, \mathbf{y}_2 \in \mathbb{R}^{d_2}$, $\mathbf{x}_3, \mathbf{y}_3 \in \mathbb{R}^{d_3}$ such that

$$A = \mathbf{x}_1\otimes\mathbf{x}_2\otimes\mathbf{y}_3 + \mathbf{x}_1\otimes\mathbf{y}_2\otimes\mathbf{x}_3 + \mathbf{y}_1\otimes\mathbf{x}_2\otimes\mathbf{x}_3. \tag{5.1}$$

The proof of this theorem will occupy the next few subsections.

5.1. Reduction. Our first step is to show that we can limit our attention to the particular tensor space $\mathbb{R}^{2\times2\times2}$. Here the orthogonal group action is important. Recall that the actions of $\mathrm{O}_{d_1,\dots,d_k}(\mathbb{R})$ and $\mathrm{GL}_{d_1,\dots,d_k}(\mathbb{R})$ on $\mathbb{R}^{d_1\times\cdots\times d_k}$ are continuous and carry decomposable tensors to decomposable tensors. It follows that the subspaces $\mathcal{S}_r$ and $\overline{\mathcal{S}}_r$ are preserved. The next theorem provides a general mechanism for passing to a tensor subspace.

THEOREM 5.2. *Let $r_i = \min(r, d_i)$ for all i. The restricted maps*

$$\mathrm{O}_{d_1,\dots,d_k}(\mathbb{R}) \times \mathcal{S}_r(r_1,\dots,r_k) \to \mathcal{S}_r(d_1,\dots,d_k),$$
$$\mathrm{O}_{d_1,\dots,d_k}(\mathbb{R}) \times \overline{\mathcal{S}}_r(r_1,\dots,r_k) \to \overline{\mathcal{S}}_r(d_1,\dots,d_k)$$

given by $((L_1,\dots,L_k), A) \mapsto (L_1,\dots,L_k)\cdot A$ are both surjective.

In other words, every rank-r tensor in $\mathbb{R}^{d_1\times\cdots\times d_k}$ is equivalent by an orthogonal transformation to a rank-r tensor in the smaller space $\mathbb{R}^{r_1\times\cdots\times r_k}$. Similarly every rank-$r$-approximable tensor in $\mathbb{R}^{d_1\times\cdots\times d_k}$ is equivalent to a rank-r-approximable tensor in $\mathbb{R}^{r_1\times\cdots\times r_k}$.

Proof. If $A \in \mathcal{S}_r(d_1,\dots,d_k)$ is any rank-r tensor then we can write $A = \sum_{j=1}^r \mathbf{x}_1^j \otimes \cdots \otimes \mathbf{x}_k^j$ for vectors $\mathbf{x}_i^j \in \mathbb{R}^{d_i}$. For each i, the vectors $\mathbf{x}_i^1,\dots,\mathbf{x}_i^r$ span a subspace $V_i \subset \mathbb{R}^{d_i}$ of rank at most r_i. Choose $L_i \in \mathrm{O}_{d_i}(\mathbb{R})$ so that $L_i(\mathbb{R}^{d_i}) \supseteq V_i$. Let $B = (L_1^{-1},\dots,L_k^{-1})\cdot A$. Then $A = (L_1,\dots,L_k)\cdot B$ and $B \in \mathcal{S}_r(d_1,\dots,d_k)$. This argument shows that the first of the maps is surjective.

Now let $A \in \overline{\mathcal{S}}_r(d_1, \dots, d_k)$ be any rank-r-approximable tensor. Let $(A^{(n)})_{n=1}^{\infty}$ be any sequence of rank-r tensors converging to A. For each n, by the preceding result, we can find $B^{(n)} \in \overline{\mathcal{S}}_r(d_1, \dots, d_k)$ and $(L_1^{(n)}, \dots, L_k^{(n)}) \in \mathrm{O}_{d_1,\dots,d_k}(\mathbb{R})$ with $(L_1^{(n)}, \dots, L_k^{(n)}) \cdot B^{(n)} = A^{(n)}$. Since $\mathrm{O}_{d_1,\dots,d_k}(\mathbb{R})$ is compact, there is a convergent subsequence $(L_1^{(n_j)}, \dots, L_k^{(n_j)}) \to (L_1, \dots, L_k)$. Let $B = (L_1, \dots, L_k)^{-1} \cdot A$. Then $A = (L_1, \dots, L_k) \cdot B$; and $B^{(n_j)} = (L_1^{(n_j)}, \dots, L_k^{(n_j)})^{-1} \cdot A^{(n_j)} \to (L_1, \dots, L_k)^{-1} \cdot A = B$, so $B \in \overline{\mathcal{S}}_r(d_1, \dots, d_k)$. Thus the second map is also surjective. ∎

COROLLARY 5.3. *If Theorem* 5.1 *is true for the tensor space* $\mathbb{R}^{2\times 2\times 2}$ *then it is true in general.*

Proof. The general case is $V_1 \otimes V_2 \otimes V_3 \cong \mathbb{R}^{d_1 \times d_2 \times d_3}$. Suppose $A \in \overline{\mathcal{S}}_2(d_1, d_2, d_3)$ and $\operatorname{rank}_{\otimes}(A) \geq 3$. By Theorem 5.2, there exists $(L_1, L_2, L_3) \in \mathrm{O}_{d_1,d_2,d_3}(\mathbb{R})$ and $B \in \overline{\mathcal{S}}_2(2, 2, 2)$ with $(L_1, L_2, L_3) \cdot B = A$. Moreover, $\operatorname{rank}_{\otimes}(B) = \operatorname{rank}_{\otimes}(A) \geq 3$ in $\mathbb{R}^{l\times m\times n}$ and hence $\operatorname{rank}_{\otimes}(B) \geq 3$ in $\mathbb{R}^{2\times 2\times 2}$ by Proposition 3.1. Since the theorem is assumed true for $\mathbb{R}^{2\times 2\times 2}$ and B satisfies the hypotheses, it can be written in the specified form in terms of vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$. It follows that A takes the same form with respect to the vectors $L_1\mathbf{x}_1, L_2\mathbf{x}_2, L_3\mathbf{x}_3$ and $L_1\mathbf{y}_1, L_2\mathbf{y}_2, L_3\mathbf{y}_3$. ∎

5.2. Tensors of rank 1 and 2. We establish two simple facts for later use.

PROPOSITION 5.4. *If* $A \in \mathbb{R}^{d_1\times\cdots\times d_k}$ *has rank* 1, *then we can write* $A = (L_1, \dots, L_k) \cdot B$, *where* $(L_1, \dots, L_k) \in \mathrm{GL}_{d_1,\dots,d_k}(\mathbb{R})$ *and* $B = \mathbf{e}_1 \otimes \cdots \otimes \mathbf{e}_k$.

Proof. Write $A = \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_k$ and choose the L_i so that $L_i(\mathbf{e}_i) = \mathbf{x}_i$. ∎

PROPOSITION 5.5. *Assume* $d_i \geq 2$ *for all* i. *If* $A \in \mathbb{R}^{d_1\times\cdots\times d_k}$ *has rank* 2, *then we can write* $A = (L_1, \dots, L_k) \cdot B$, *where* $(L_1, \dots, L_k) \in \mathrm{GL}_{d_1,\dots,d_k}(\mathbb{R})$ *and* $B \in \mathbb{R}^{2\times\cdots\times 2}$ *is of the form* $B = \mathbf{e}_1 \otimes \cdots \otimes \mathbf{e}_1 + \mathbf{f}_1 \otimes \cdots \otimes \mathbf{f}_k$. *Here* $\mathbf{e}_1$ *denotes the standard basis vector* $(1, 0)^\top$; *each* $\mathbf{f}_i$ *is equal either to* $\mathbf{e}_1$ *or to* $\mathbf{e}_2 = (0, 1)^\top$; *and at least two of the* $\mathbf{f}_i$ *are equal to* $\mathbf{e}_2$.

Proof. We can write $A = \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_k + \mathbf{y}_1 \otimes \cdots \otimes \mathbf{y}_k$. Since $\operatorname{rank}_{\otimes}(A) = 2$ all of the $\mathbf{x}_i$ and $\mathbf{y}_i$ must be nonzero. We claim that $\mathbf{y}_i, \mathbf{x}_i$ must be linearly independent for at least two different indices i. Otherwise, suppose $\mathbf{y}_i = \lambda_i\mathbf{x}_i$ for $k - 1$ different indices, say, $i = 1, \dots, k - 1$. It would follow that

$$A = \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_{k-1} \otimes \ (\mathbf{x}_k + (\lambda_1 \cdots \lambda_{k-1})\mathbf{y}_k),$$

contradicting $\operatorname{rank}_{\otimes}(A) = 2$.

For each i choose $L_i : \mathbb{R}^2 \to \mathbb{R}^{d_i}$ such that $L_i\mathbf{e}_1 = \mathbf{x}_i$ and such that $L_i\mathbf{e}_2 = \mathbf{y}_i$ if $\mathbf{y}_i$ is linearly independent of $\mathbf{x}_i$; otherwise $L_i\mathbf{e}_2$ may be arbitrary. It is easy to check that $(L_1, \dots, L_k)^{-1} \cdot A = \mathbf{e}_1 \otimes \cdots \otimes \mathbf{e}_1 + \lambda\mathbf{f}_1 \otimes \cdots \otimes \mathbf{f}_k$, where the $\mathbf{f}_i$ are as specified in the theorem, and λ is the product of the λ_i over those indices where $\mathbf{y}_i = \lambda_i\mathbf{x}_i$. This is almost in the correct form. To get rid of the λ, replace $L_i\mathbf{e}_2 = \mathbf{y}_i$ with $L_i\mathbf{e}_2 = \lambda\mathbf{y}_i$ at one of the indices i for which $\mathbf{x}_i, \mathbf{y}_i$ are linearly independent. This completes the construction. ∎

5.3. The discriminant polynomial Δ. The structure of tensors in $\mathbb{R}^{2\times 2\times 2}$ is largely governed by a quartic polynomial Δ which we define and discuss here. This same polynomial was discovered by Cayley in 1845 [15]. More generally, Δ is the $2 \times 2 \times 2$ special case of an object called the *hyperdeterminant* revived in its modern form by Gelfand, Kapranov, and Zelevinsky [30, 31]. We give an elementary treatment of the properties we need.

As in our discussion in section 2.1, we identify a tensor $\mathbf{A} \in \mathbb{R}^2 \otimes \mathbb{R}^2 \otimes \mathbb{R}^2$ with the array $A \in \mathbb{R}^{2\times 2\times 2}$ of its eight coefficients with respect to the standard basis

$\{\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k : i, j, k = 1, 2\}$. Pictorially, we can represent it as a pair of side-by-side 2×2 slabs:

$$\mathbf{A} = \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} a_{ijk}\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k = \left[\begin{array}{cc|cc} a_{111} & a_{112} & a_{211} & a_{212} \\ a_{121} & a_{122} & a_{221} & a_{222} \end{array}\right] = A.$$

The general strategy is to find ways of simplifying the representation of A by applying transformations in $\mathrm{GL}_{2,2,2}(\mathbb{R}) = \mathrm{GL}_2(\mathbb{R}) \times \mathrm{GL}_2(\mathbb{R}) \times \mathrm{GL}_2(\mathbb{R})$. This group is generated by the following operations: decomposable row operations applied to both slabs simultaneously; decomposable column operations applied to both slabs simultaneously; decomposable slab operations (for example, adding a multiple of one slab to the other).

Slab operations on a tensor $A = [A_1 \,|\, A_2]$ generate new 2×2 slabs of the form $S = \lambda_1 A_1 + \lambda_2 A_2$. One can check that

$$\det(S) = \lambda_1^2 \det(A_1) + \lambda_1\lambda_2 \frac{\det(A_1 + A_2) - \det(A_1 - A_2)}{2} + \lambda_2^2 \det(A_2). \tag{5.2}$$

We define Δ to be the discriminant of this quadratic polynomial:

$$\Delta([A_1 \,|\, A_2]) = \left[\frac{\det(A_1 + A_2) - \det(A_1 - A_2)}{2}\right]^2 - 4\det(A_1)\det(A_2). \tag{5.3}$$

Explicitly, if $A = [\![a_{ijk}]\!]_{i,j,k=1,2} \in \mathbb{R}^{2\times2\times2}$, then

$$\begin{aligned}\Delta(A) = {} & (a_{111}^2a_{222}^2 + a_{112}^2a_{221}^2 + a_{121}^2a_{212}^2 + a_{122}^2a_{211}^2) \\ & - 2(a_{111}a_{112}a_{221}a_{222} + a_{111}a_{121}a_{212}a_{222} + a_{111}a_{122}a_{211}a_{222} \\ & + a_{112}a_{121}a_{212}a_{221} + a_{112}a_{122}a_{221}a_{211} + a_{121}a_{122}a_{212}a_{211}) \\ & + 4(a_{111}a_{122}a_{212}a_{221} + a_{112}a_{121}a_{211}a_{222}).\end{aligned}$$

PROPOSITION 5.6. *Let $A \in \mathbb{R}^{2\times2\times2}$, let A' be obtained from A by permuting the three factors in the tensor product, and let $(L_1, L_2, L_3) \in \mathrm{GL}_{2,2,2}(\mathbb{R})$. Then $\Delta(A') = \Delta(A)$ and $\Delta((L_1, L_2, L_3) \cdot A) = \det(L_1)^2 \det(L_2)^2 \det(L_3)^2 \Delta(A)$.*

Proof. To show that Δ is invariant under all permutations of the factors of $\mathbb{R}^{2\times2\times2}$, it is enough to check invariance in the cases of two distinct transpositions. It is clear from (5.3) that Δ is invariant under the transposition of the second and third factors, since this amounts to replacing A_1, A_2 with their transposes $A_1^\top, A_2^\top$. To show that Δ is invariant under transposition of the first and third factors, write $A = [\mathbf{u}_{11}, \mathbf{u}_{12} \,|\, \mathbf{u}_{21}, \mathbf{u}_{22}]$, where the $\mathbf{u}_{ij}$ are column vectors. One can verify that

$$\begin{aligned}\Delta(A) = {} & \det[\mathbf{u}_{11}, \mathbf{u}_{22}]^2 + \det[\mathbf{u}_{21}, \mathbf{u}_{12}]^2 \\ & - 2\det[\mathbf{u}_{11}, \mathbf{u}_{12}]\det[\mathbf{u}_{21}, \mathbf{u}_{22}] - 2\det[\mathbf{u}_{11}, \mathbf{u}_{21}]\det[\mathbf{u}_{12}, \mathbf{u}_{22}],\end{aligned}$$

which has the desired symmetry.

In view of the permutation invariance of Δ, it is enough to verify the second claim in the case $(L_1, L_2, L_3) = (I, L_2, I)$. Then $(L_1, L_2, L_3) \cdot A = [L_2A_1 \,|\, L_2A_2]$ and an extra factor $\det(L_2)^2$ appears in all terms of (5.3), exactly as required. ∎

COROLLARY 5.7. *The sign of Δ is invariant under the action of $\mathrm{GL}_{2,2,2}(\mathbb{R})$.*

COROLLARY 5.8. *The value of Δ is invariant under the action of $\mathrm{O}_{2,2,2}(\mathbb{R})$.*

Using the properties of Δ, we can easily prove, in a slightly different way, a result due originally to Kruskal (unpublished work) and ten Berge [73].

PROPOSITION 5.9. *If* $\Delta(A) > 0$ *then* $\operatorname{rank}_\otimes(A) \leq 2$.

PROPOSITION 5.10. *If* $\operatorname{rank}_\otimes(A) \leq 2$ *then* $\Delta(A) \geq 0$.

Proof of Proposition 5.9. If the discriminant $\Delta(A)$ is positive then the homogeneous quadratic equation (5.2) has two linearly independent root pairs $(\lambda_{11}, \lambda_{12})$ and $(\lambda_{21}, \lambda_{22})$. It follows that we can use slab operations to transform $[A_1 \mid A_2] \to [B_1 \mid B_2]$, where $B_i = \lambda_{i1}A_1 + \lambda_{i2}A_2$. By construction $\det(B_i) = 0$, so we can write $B_i = \mathbf{f}_i \otimes \mathbf{g}_i$ for some $\mathbf{f}_i, \mathbf{g}_i \in \mathbb{R}^2$ (possibly zero). It follows that $[B_1 \mid B_2] = \mathbf{e}_1 \otimes \mathbf{f}_1 \otimes \mathbf{g}_1 + \mathbf{e}_2 \otimes \mathbf{f}_2 \otimes \mathbf{g}_2$; so $\operatorname{rank}_\otimes(A) = \operatorname{rank}_\otimes([B_1 \mid B_2]) \leq 2$. ∎

Proof of Proposition 5.10. It is easy to check that $\Delta(A) = 0$ if $\operatorname{rank}_\otimes(A) \leq 1$, since we can write $A = (L_1, L_2, L_3) \cdot (\mathbf{e}_1 \otimes \mathbf{e}_1 \otimes \mathbf{e}_1)$ or else $A = 0$.

It remains to be shown that $\Delta(A)$ is not negative when $\operatorname{rank}_\otimes(A) = 2$. Proposition 5.5 implies that A can be transformed by an element of $\mathrm{GL}_{2,2,2}(\mathbb{R})$ (and a permutation of factors, if necessary) into one of the following tensors:

$$I_1 = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \quad \text{or} \quad I_2 = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right].$$

Since $\Delta(I_1) = 1$ and $\Delta(I_2) = 0$, it follows that $\Delta(A) \geq 0$. ∎

Kruskal and also ten Berge deserve complete credit for discovering the above result. In fact, the hyperdeterminant for $2 \times 2 \times 2$ tensor Δ is known by the name *Kruskal polynomial* in the psychometrics community [73]. Our goal is not so much to provide alternative proofs for Propositions 5.9 and 5.10 but to include them so that our proof of Theorem 5.1 can be self-contained. We are now ready to give that proof, thereby characterizing all limit points of order-3 rank-2 tensors.

Proof of Theorem 5.1. Note that the theorem is stated for order-3 tensors of any size $d_1 \times d_2 \times d_3$. We begin with the case $A \in \mathbb{R}^{2\times 2\times 2}$. Suppose $A \in \overline{\mathcal{S}}_2(2,2,2) \setminus \mathcal{S}_2(2,2,2)$. Then we claim that $\Delta(A) = 0$. Indeed, since $A \notin \mathcal{S}_2$, Proposition 5.9 implies that $\Delta(A) \leq 0$. On the other hand, since $A \in \overline{\mathcal{S}}_2$, it follows from Proposition 5.10 and the continuity of Δ that $\Delta(A) \geq 0$.

Since $\Delta(A) = 0$, the homogeneous quadratic equation (5.2) has a nontrivial root pair (λ_1, λ_2). It follows that A can be transformed by slab operations into the form $[A_i \mid S]$, where $S = \lambda_1 A_1 + \lambda_2 A_2$ and $i = 1$ or 2. By construction $\det(S) = 0$, but $S \neq 0$ for otherwise $\operatorname{rank}_\otimes(A) = \operatorname{rank}(A_i) \leq 2$. Hence $\operatorname{rank}(S) = 1$ and by a further transformation we can reduce A to the form

$$B = \left[\begin{array}{cc|cc} p & q & 1 & 0 \\ r & s & 0 & 0 \end{array}\right].$$

In fact we may assume $p = 0$ (the operation "subtract p times the second slab from the first slab" will achieve this), and moreover $s^2 = \Delta(B) = 0$. Both q and r must be nonzero; otherwise $\operatorname{rank}_\otimes(A) = \operatorname{rank}_\otimes(B) \leq 2$. If we rescale the bottom rows by $1/r$ and the right-hand columns by $1/q$ we are finally reduced to

$$B' = \left[\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{array}\right] = \mathbf{e}_2 \otimes \mathbf{e}_1 \otimes \mathbf{e}_1 + \mathbf{e}_1 \otimes \mathbf{e}_2 \otimes \mathbf{e}_1 + \mathbf{e}_1 \otimes \mathbf{e}_1 \otimes \mathbf{e}_2.$$

By reversing all the row, column, and slab operations we can obtain a transformation $(L_1, L_2, L_3) \in \mathrm{GL}_{2,2,2}(\mathbb{R})$ such that $A = (L_1, L_2, L_3) \cdot B'$. Then A can be written in the required form, with $\mathbf{x}_i = L_i\mathbf{e}_1$, $\mathbf{y}_i = L_i\mathbf{e}_2$ for $i = 1, 2, 3$.

This completes the proof of Theorem 5.1 in the case of the tensor space $\mathbb{R}^{2\times 2\times 2}$. By Corollary 5.3 this implies the theorem in general. ∎

5.4. Ill-posedness and ill-conditioning of the best rank-r approximation problem. Recall that a problem is called *well-posed* if a solution exists, is unique, and is stable (i.e., depends continuously on the input data). If one or more of these three criteria are not satisfied, the problem[3] is called *ill-posed*.

From sections 4 and 8, we see that tensors will often fail to have a best rank-r approximation. In all applications that rely on $\text{APPROX}(A, r)$ or a variant of it as the underlying mathematical model, we should fully expect the ill-posedness of $\text{APPROX}(A, r)$ to pose a serious difficulty. Even if it is known a priori that a tensor A has a best rank-r approximation, we should remember that in applications, the data array $\hat{A}$ available at our disposal is almost always one that is corrupted by noise, i.e., $\hat{A} = A + E$, where E denotes the collective contributions of various errors, limitations in measurements, background noise, rounding off, etc. Clearly there is no guarantee that $\hat{A}$ will also have a best rank-r approximation.

In many situations, one needs only a "good" rank-r approximation rather than the best rank-r approximation. It is tempting to argue, then, that the nonexistence of the best solution does not matter—it is enough to seek an "approximate solution." We discourage this point of view for two main reasons. First, there is a serious conceptual difficulty: if there is no solution, then what is the "approximate solution" an approximation of? Second, even if one disregards this and ploughs ahead to compute an "approximate solution," we argue below that this task is ill-conditioned and the computation is unstable.

For notational simplicity and since there is no loss of generality (cf. Theorem 4.10 and Corollary 4.12), we will use the problem of finding a best rank-2 approximation to a rank-3 tensor to make our point. Let $A \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ be an instance where

$$\operatorname{argmin}_{\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^{d_i}} \|A - \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 - \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3\| \tag{5.4}$$

does not have a solution (such examples abound; cf. section 8). If we disregard the fact that a solution does not exist and plug the problem into a computer program,[4] we will still get some sort of "approximate solution" because of the finite-precision error inherent in the computer. What really happens here [77] is that we are effectively solving a problem perturbed by some small $\varepsilon > 0$; the "approximate solution" $\mathbf{x}_i^*(\varepsilon), \mathbf{y}_i^*(\varepsilon) \in \mathbb{R}^{d_i}$ $(i = 1, 2, 3)$ is really a solution to the perturbed problem

$$\begin{aligned} &\|A - \mathbf{x}_1^*(\varepsilon) \otimes \mathbf{x}_2^*(\varepsilon) \otimes \mathbf{x}_3^*(\varepsilon) - \mathbf{y}_1^*(\varepsilon) \otimes \mathbf{y}_2^*(\varepsilon) \otimes \mathbf{y}_3^*(\varepsilon)\| \\ &\qquad = \varepsilon + \inf_{\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^{d_i}} \|A - \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 - \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3\|. \end{aligned} \tag{5.5}$$

Since we are attempting to find a solution of (5.4) that does not exist, in exact arithmetic the algorithm will never terminate, but in reality the computer is limited by its finite precision, and so the algorithm terminates at an "approximate solution," which may be viewed as a solution to a perturbed problem (5.5). This process of forcing a solution to an ill-posed problem is almost always guaranteed to be ill-conditioned because of the infamous rule of thumb in numerical analysis [22, 23, 24]:

A well-posed problem near an ill-posed one is ill-conditioned.

[3] Normally, existence is taken for granted, and an ill-posed problem often means one whose solution lacks either uniqueness or stability. In this paper, the ill-posedness is of a more serious kind—the existence of a solution is itself in question.

[4] While there is no known globally convergent algorithm for $\text{APPROX}(A, r)$, we will ignore this difficulty for a moment and assume that the ubiquitous alternating least squares algorithm would yield the required solution.

The root of the ill-conditioning lies in the fact that we are solving the (well-posed but ill-conditioned) problem (5.5) that is a slight perturbation of the ill-posed problem (5.4). The ill-conditioning manifests itself as the phenomenon described in Proposition 4.8, namely,

$$\|\mathbf{x}_1^*(\varepsilon) \otimes \mathbf{x}_2^*(\varepsilon) \otimes \mathbf{x}_3^*(\varepsilon)\| \to \infty \qquad \text{and} \qquad \|\mathbf{y}_1^*(\varepsilon) \otimes \mathbf{y}_2^*(\varepsilon) \otimes \mathbf{y}_3^*(\varepsilon)\| \to \infty$$

as $\varepsilon \to 0$. The ill-conditioning described here was originally observed in numerical experiments by psychometricians and chemometricians, who named the phenomenon "diverging CANDECOMP/PARAFAC components" or "CANDECOMP/PARAFAC degeneracy" [49, 51, 63, 68, 69].

To fix the ill-conditioning, we should first fix the ill-posedness, i.e., find a well-posed problem. This leads us to the subject of the next section.

5.5. Weak solutions. In the study of PDEs [29], there often arise systems of PDEs that have no solutions in the traditional sense. A standard way around this is to define a so-called *weak solution*, which may not be a continuous function or even a function (which is a tad odd since one would expect a *solution* to a PDE to be at least differentiable). Without going into the details, we will just say that weak solution turns out to be an extremely useful concept and is indispensable in modern studies of PDEs. Under the proper context, a weak solution to an ill-posed PDE may be viewed as the limit of *strong* or *classical solutions* to a sequence of well-posed PDEs that are slightly perturbed versions of the ill-posed one in question. Motivated by the PDE analogies, we will define weak solutions to APPROX(A, r).

We let $\mathcal{S}_r(d_1, \dots, d_k) := \{A \in \mathbb{R}^{d_1 \times \cdots \times d_k} \mid \operatorname{rank}_\otimes(A) \le r\}$ and let $\overline{\mathcal{S}}_r(d_1, \dots, d_k)$ denote its closure in the (unique) norm-topology.

DEFINITION 5.11. *An order-k tensor $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ has* border rank r *if*

$$A \in \overline{\mathcal{S}}_r(d_1, \dots, d_k) \qquad \textit{and} \qquad A \notin \overline{\mathcal{S}}_{r-1}(d_1, \dots, d_k).$$

This is denoted by $\underline{\operatorname{rank}}_\otimes(A)$. *Note that*

$$\overline{\mathcal{S}}_r(d_1, \dots, d_k) = \{A \in \mathbb{R}^{d_1 \times \cdots \times d_k} \mid \underline{\operatorname{rank}}_\otimes(S) \le r\}.$$

Remark. Clearly $\underline{\operatorname{rank}}_\otimes(A) \le \operatorname{rank}_\otimes(A)$ for any tensor A. Since $\overline{\mathcal{S}}_0 = \mathcal{S}_0$ (trivially) and $\overline{\mathcal{S}}_1 = \mathcal{S}_1$ (by Proposition 4.2), it follows that $\underline{\operatorname{rank}}_\otimes(A) = \operatorname{rank}_\otimes(A)$ whenever $\operatorname{rank}_\otimes(A) \le 2$. Moreover, $\underline{\operatorname{rank}}_\otimes(A) \ge 2$ if $\operatorname{rank}_\otimes(A) \ge 2$.

Our definition differs slightly from the usual definition of border rank in the algebraic computational complexity literature [5, 6, 12, 48, 54], which uses the Zariski topology (and is normally defined for tensors over $\mathbb{C}$).

Let $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ with $d_i \ge 2$ and $k \ge 3$. Then the way to ensure that APPROX(A, r), the optimal rank-r approximation problem

$$\operatorname{argmin}_{\operatorname{rank}_\otimes(B) \le r} \|A - B\|, \tag{5.6}$$

always has a meaningful solution for any $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ is to instead consider the optimal border-rank-r approximation problem

$$\operatorname{argmin}_{\underline{\operatorname{rank}}_\otimes(B) \le r} \|A - B\|. \tag{5.7}$$

It is an obvious move to propose to fix the ill-posedness of APPROX(A, r) by taking the closure. However, without a characterization of the limit points such a proposal

will at best be academic—it is not enough to simply say that weak solutions are limits of rank-2 tensors without giving an explicit expression (or a number of expressions) for them that may be plugged into the objective function to be minimized.

Theorem 5.1 solves this problem in the order-3 rank-2 case—it gives a complete description of these limit points with an explicit formula and, in turn, a constructive solution to the border-rank approximation problem. In case this is not obvious, we will spell out the implication of Theorem 5.1.

COROLLARY 5.12. *Let* $d_1, d_2, d_3 \geq 2$. *Let* $A \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ *with* $\operatorname{rank}_{\otimes}(A) = 3$. A *is the limit of a sequence* $A_n \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ *with* $\operatorname{rank}_{\otimes}(A_n) \leq 2$ *if and only if*

$$A = \mathbf{y}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 + \mathbf{x}_1 \otimes \mathbf{y}_2 \otimes \mathbf{x}_3 + \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{y}_3$$

for some $\mathbf{x}_i, \mathbf{y}_i$ *linearly independent vectors in* $\mathbb{R}^{d_i}$, $i = 1, 2, 3$.

This implies that every tensor in $\overline{\mathcal{S}}_2(d_1, \dots, d_k)$ can be written in one of two forms:

$$\mathbf{y}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 + \mathbf{x}_1 \otimes \mathbf{y}_2 \otimes \mathbf{x}_3 + \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{y}_3 \tag{5.8}$$

or

$$\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 + \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3. \tag{5.9}$$

These expressions may then be used to define the relevant objective function(s) in the minimization of (5.7). As in the case of PDEs, every classical (strong) solution is also a weak solution to APPROX(A, r).

PROPOSITION 5.13. *If* B *is a solution to* (5.6) *then* B *is a solution to* (5.7).

Proof. If $\|A - B\| \leq \|A - B'\|$ for all $B' \in \mathcal{S}_r$, then $\|A - B\| \leq \|A - B'\|$ for all $B' \in \overline{\mathcal{S}}_r$ by continuity. ∎

6. Semialgebraic description of tensor rank. One may wonder whether the result in Propositions 5.9 and 5.10 extends to more general *hyperdeterminants*. We know from [30, 31] that a hyperdeterminant may be uniquely defined (up to a constant scaling) in $\mathbb{R}^{d_1 \times \cdots \times d_k}$ whenever $d_1, \dots, d_k$ satisfy

$$d_i - 1 \leq \sum_{j \neq i} (d_j - 1) \qquad \text{for } i = 1, \dots, k. \tag{6.1}$$

(Note that for matrices, (6.1) translates to $d_1 = d_2$, which may be viewed as one reason why the determinant is defined only for square matrices.) Let $\operatorname{Det}_{d_1,\dots,d_k} : \mathbb{R}^{d_1 \times \cdots \times d_k} \to \mathbb{R}$ be the polynomial function defined by the hyperdeterminant whenever (6.1) is satisfied. Propositions 5.9 and 5.10 tell us that the rank of a tensor is 2 on the set $\{A \mid \operatorname{Det}_{2,2,2}(A) > 0\}$ and 3 on the set $\{A \mid \operatorname{Det}_{2,2,2}(A) < 0\}$. One may start by asking whether the tensor rank in $\mathbb{R}^{d_1 \times \cdots \times d_k}$ is constant-valued on the sets

$$\{A \mid \operatorname{Det}_{d_1,\dots,d_k}(A) < 0\} \qquad \text{and} \qquad \{A \mid \operatorname{Det}_{d_1,\dots,d_k}(A) > 0\}.$$

The answer, as Sturmfels has kindly communicated to us [71], is *no* with explicit counterexamples in cases $2 \times 2 \times 2 \times 2$ and $3 \times 3 \times 3$. We will not reproduce Sturmfels's examples here (one reason is that $\operatorname{Det}_{2,2,2,2}$ already contains close to 3 million monomial terms [35]) but instead refer our readers to his forthcoming paper.

We will prove that although there is no single polynomial Δ that will separate $\mathbb{R}^{d_1 \times \cdots \times d_k}$ into regions of constant rank as in the case of $\mathbb{R}^{2 \times 2 \times 2}$, there is always a finite number of polynomials $\Delta_1, \dots, \Delta_m$ that will achieve this.

Before we state and prove the result, we will introduce a few notions and notations. We will write $\mathbb{R}[X_1, \dots, X_m]$ for the ring of polynomials in m variables $X_1, \dots, X_m$ with real coefficients. Subsequently, we will be considering polynomial functions on tensor spaces and will index our variables in a consistent way (for example, when discussing polynomial functions on $\mathbb{R}^{l\times m\times n}$, the polynomial ring in question will be denoted $\mathbb{R}[X_{111}, X_{112}, \dots X_{lmn}]$). Given $A = [\![a_{ijk}]\!] \in \mathbb{R}^{l\times m\times n}$ and $p(X_{111}, X_{112}, \dots X_{lmn}) \in \mathbb{R}[X_{111}, X_{112}, \dots X_{lmn}]$, $p(A)$ will mean the obvious thing, namely, $p(A) = p(a_{111}, a_{112}, \dots, a_{lmn}) \in \mathbb{R}$.

A *polynomial map* is a function $F : \mathbb{R}^n \to \mathbb{R}^m$, defined for each $\mathbf{a} = [a_1, \dots, a_n]^\top \in \mathbb{R}^n$ by $F(\mathbf{a}) = [f_1(\mathbf{a}), \dots, f_m(\mathbf{a})]^\top$, where $f_i \in \mathbb{R}[X_1, \dots, X_n]$ for all $i = 1, \dots, m$.

A *semialgebraic set* in $\mathbb{R}^n$ is a union of finitely many sets of the form[5]

$$\{\mathbf{a} \in \mathbb{R}^n \mid p(\mathbf{a}) = 0,\ q_1(\mathbf{a}) > 0,\ \dots,\ q_\ell(\mathbf{a}) > 0\},$$

where $\ell \in \mathbb{N}$ and $p, q_1, \dots, q_\ell \in \mathbb{R}[X_1, \dots, X_n]$. Note that we do not exclude the possibility of p or any of the q_i being constant (degree-0) polynomials. For example, if p is the zero polynomial, then the first relation $0 = 0$ is trivially satisfied and the semialgebraic set will be an open set in $\mathbb{R}^n$.

It is easy to see that the class of all semialgebraic sets in $\mathbb{R}^n$ is closed under finite unions, finite intersections, and taking the complement. Moreover, if $\mathcal{S} \subseteq \mathbb{R}^{n+1}$ is a semialgebraic set and $\pi : \mathbb{R}^{n+1} \to \mathbb{R}^n$ is the projection onto the first n coordinates, then $\pi(\mathcal{S})$ is also a semialgebraic set; this seemingly innocuous statement is in fact the Tarski–Seidenberg theorem [65, 72], possibly the most celebrated result about semialgebraic sets. We will restate it in a (somewhat less common) form that better suits our purpose.

THEOREM 6.1 (Tarski–Seidenberg). *If $\mathcal{S} \subseteq \mathbb{R}^n$ is a semialgebraic set and $F : \mathbb{R}^n \to \mathbb{R}^m$ is a polynomial map, then the image $F(\mathcal{S}) \subseteq \mathbb{R}^m$ is also a semialgebraic set.*

These and other results about semialgebraic sets may be found in [19, Chapter 2], which, in addition, is a very readable introduction to semialgebraic geometry.

THEOREM 6.2. *The set $\mathcal{R}_r(d_1, \dots, d_k) := \{A \in \mathbb{R}^{d_1\times\cdots\times d_k} \mid \operatorname{rank}_\otimes(A) = r\}$ is a semialgebraic set.*

Proof. Let $\psi_r : (\mathbb{R}^{d_1} \times \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_k})^r \to \mathbb{R}^{d_1\times d_2\times\cdots\times d_k}$ be defined by

$$\psi_r(\mathbf{u}_1, \mathbf{v}_1, \dots, \mathbf{z}_1; \dots; \mathbf{u}_r, \mathbf{v}_r, \dots, \mathbf{z}_r) = \mathbf{u}_1 \otimes \mathbf{v}_1 \otimes \cdots \otimes \mathbf{z}_1 + \cdots + \mathbf{u}_r \otimes \mathbf{v}_r \otimes \cdots \otimes \mathbf{z}_r.$$

It is clear that the image of ψ_r is exactly $\mathcal{S}_r(d_1, \dots, d_k) = \{A \mid \operatorname{rank}_\otimes(A) \le r\}$. It is also clear that ψ_r is a polynomial map.

It follows from Theorem 6.1 that $\mathcal{S}_r(d_1, \dots, d_k)$ is semialgebraic. This holds for arbitrary r. So $\mathcal{R}_r(d_1, \dots, d_k) = \mathcal{S}_r(d_1, \dots, d_k) \setminus \mathcal{S}_{r-1}(d_1, \dots, d_k)$ is also semialgebraic. ∎

COROLLARY 6.3. *There exist $\Delta_0, \dots, \Delta_m \in \mathbb{R}[X_{1\cdots 1}, \dots, X_{d_1\cdots d_k}]$ from which the rank of a tensor $A \in \mathbb{R}^{d_1\times\cdots\times d_k}$ can be determined purely from the signs (i.e., + or − or 0) of $\Delta_0(A), \dots, \Delta_m(A)$.*

In the next section, we will see examples of such polynomials for the tensor space $\mathbb{R}^{2\times 2\times 2}$. We will stop short of giving an explicit semialgebraic characterization of rank, but it should be clear to the reader how to get one.

[5] Only one p is necessary, because multiple equality constraints $p_1(\mathbf{a}) = 0, \dots, p_k(\mathbf{a}) = 0$ can always be amalgamated into a single equation $p(\mathbf{a}) = 0$ by setting $p = p_1^2 + \cdots + p_k^2$.

7. Orbits of real $2 \times 2 \times 2$ tensors. In this section, we study the equivalence of tensors in $\mathbb{R}^{2\times2\times2}$ under multilinear matrix multiplication. We will use the results and techniques of this section later on in section 8 where we determine *which* tensors in $\mathbb{R}^{2\times2\times2}$ have an optimal rank-2 approximation.

Recall that A and $B \in \mathbb{R}^{2\times2\times2}$ are said to be ($\mathrm{GL}_{2,2,2}(\mathbb{R})$-)equivalent if and only if there exists a transformation $(L, M, N) \in \mathrm{GL}_{2,2,2}(\mathbb{R})$ such that $A = (L, M, N) \cdot B$. The question is whether there is a finite list of "canonical tensors" so that every $A \in \mathbb{R}^{2\times2\times2}$ is equivalent to one of them. For matrices $A \in \mathbb{R}^{m\times n}$, $\operatorname{rank}(A) = r$ if and only if there exist $M \in \mathrm{GL}_m(\mathbb{R}), N \in \mathrm{GL}_n(\mathbb{R})$ such that

$$(M, N) \cdot A = MAN^{\top} = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}.$$

So every matrix of rank r is equivalent to one that takes the canonical form $\left[\begin{smallmatrix} I_r & 0 \\ 0 & 0 \end{smallmatrix}\right]$. Note that this is the same as saying that the matrix A can be transformed into $\left[\begin{smallmatrix} I_r & 0 \\ 0 & 0 \end{smallmatrix}\right]$ using elementary row and column operations—adding a scalar multiple of a row/column to another, scaling a row/column by a nonzero scalar, interchanging two rows/columns—since every $(L_1, L_2) \in \mathrm{GL}_{m,n}(\mathbb{R})$ is a sequence of such operations.

We will see that there is indeed a finite number of canonical forms for tensors in $\mathbb{R}^{2\times2\times2}$, although the classification is somewhat more intricate than the case of matrices; two tensors in $\mathbb{R}^{2\times2\times2}$ can have the same rank but be inequivalent (i.e., reduce to different canonical forms).

In fancier language, what we are doing is classifying the *orbits* of the *group action* $\mathrm{GL}_{2,2,2}(\mathbb{R})$ on $\mathbb{R}^{2\times2\times2}$. We are doing for $\mathbb{R}^{2\times2\times2}$ what Gelfand, Kapranov, and Zelevinsky did for $\mathbb{C}^{2\times2\times2}$ in the last sections of [30, 31]. Not surprisingly, the results that we obtained are similar but not identical; there are *eight* distinct orbits for the action of $\mathrm{GL}_{2,2,2}(\mathbb{R})$ on $\mathbb{R}^{2\times2\times2}$ as opposed to *seven* distinct orbits for the action of $\mathrm{GL}_{2,2,2}(\mathbb{C})$ on $\mathbb{C}^{2\times2\times2}$—a further reminder of the dependence of such results on the choice of field.

THEOREM 7.1. *Every tensor in $\mathbb{R}^{2\times2\times2}$ is equivalent via a transformation in $\mathrm{GL}_{2,2,2}(\mathbb{R})$ to precisely one of the canonical forms indicated in Table* 7.1*, with its invariants taking the values shown.*

Proof. Write $A = [A_1 \mid A_2]$, $A_i \in \mathbb{R}^{2\times2}$ for $[\![a_{ijk}]\!] \in \mathbb{R}^{2\times2\times2}$. If $\operatorname{rank}(A_1) = 0$, then

$$A = \left[\begin{array}{cc|cc} 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{array}\right].$$

Using matrix operations, A must then be equivalent to one of the forms (depending on $\operatorname{rank}(A_2)$)

$$\left[\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right], \qquad \left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right], \qquad \left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right],$$

which correspond to D_0, D_1, and D_2, respectively (after reordering the slabs).

If $\operatorname{rank}(A_1) = 1$, then we may assume that

$$A = \left[\begin{array}{cc|cc} 1 & 0 & a & b \\ 0 & 0 & c & d \end{array}\right].$$

TABLE 7.1
GL-*orbits of* $\mathbb{R}^{2\times2\times2}$. *The letters* D, G *stand for "degenerate" and "generic," respectively.*

tensor	sign(Δ)	rank	$\text{rank}_\otimes$	$\underline{\text{rank}}_\otimes$
$D_0 = \left[\begin{array}{cc\|cc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$	0	$(0,0,0)$	0	0
$D_1 = \left[\begin{array}{cc\|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$	0	$(1,1,1)$	1	1
$D_2 = \left[\begin{array}{cc\|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right]$	0	$(1,2,2)$	2	2
$D_2' = \left[\begin{array}{cc\|cc} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right]$	0	$(2,1,2)$	2	2
$D_2'' = \left[\begin{array}{cc\|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right]$	0	$(2,2,1)$	2	2
$G_2 = \left[\begin{array}{cc\|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$	+	$(2,2,2)$	2	2
$D_3 = \left[\begin{array}{cc\|cc} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right]$	0	$(2,2,2)$	3	2
$G_3 = \left[\begin{array}{cc\|cc} 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{array}\right]$	−	$(2,2,2)$	3	3

If $d \neq 0$ then we may transform this to G_2 as follows:

$$\left[\begin{array}{cc\|cc} 1 & 0 & a & b \\ 0 & 0 & c & d \end{array}\right] \rightsquigarrow \left[\begin{array}{cc\|cc} 1 & 0 & \times & 0 \\ 0 & 0 & 0 & d \end{array}\right] \rightsquigarrow \left[\begin{array}{cc\|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right].$$

If $d = 0$, then

$$\left[\begin{array}{cc\|cc} 1 & 0 & a & b \\ 0 & 0 & c & 0 \end{array}\right] \rightsquigarrow \left[\begin{array}{cc\|cc} 1 & 0 & 0 & b \\ 0 & 0 & c & 0 \end{array}\right].$$

In this situation we can normalize b, c separately, reducing these matrices to one of the following four cases (according to whether b, c are zero):

$$\left[\begin{array}{cc\|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right], \quad \left[\begin{array}{cc\|cc} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right], \quad \left[\begin{array}{cc\|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right], \quad \left[\begin{array}{cc\|cc} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right],$$

which are D_1, D_2', D_2'', and D_3, respectively.

Finally, if $\text{rank}(A_1) = 2$, then we may assume that

$$A = [A_1 \mid A_2] = \left[\begin{array}{cc\|cc} 1 & 0 & \times & \times \\ 0 & 1 & \times & \times \end{array}\right].$$

By applying a transformation of the form (I, L, L^{-1}), we can keep A_1 fixed while conjugating A_2 into (real) Jordan canonical form. There are four cases.

If A_2 has repeated real eigenvalues and is diagonalizable, then we get D_2:

$$\left[\begin{array}{cc|cc} 1 & 0 & \lambda & 0 \\ 0 & 1 & 0 & \lambda \end{array}\right] \rightsquigarrow \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right].$$

If A_2 has repeated real eigenvalues and is not diagonalizable, then we have

$$\left[\begin{array}{cc|cc} 1 & 0 & \lambda & 1 \\ 0 & 1 & 0 & \lambda \end{array}\right] \rightsquigarrow \left[\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{array}\right],$$

which is equivalent (after swapping columns and swapping slabs) to D_3.

If A_2 has distinct real eigenvalues, then A reduces to G_2:

$$\left[\begin{array}{cc|cc} 1 & 0 & \lambda & 0 \\ 0 & 1 & 0 & \mu \end{array}\right] \rightsquigarrow \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \mu-\lambda \end{array}\right] \rightsquigarrow \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right].$$

If A_2 has complex eigenvalues, then we can reduce A to G_3:

$$\left[\begin{array}{cc|cc} 1 & 0 & a & -b \\ 0 & 1 & b & a \end{array}\right] \rightsquigarrow \left[\begin{array}{cc|cc} 1 & 0 & 0 & -b \\ 0 & 1 & b & 0 \end{array}\right] \rightsquigarrow \left[\begin{array}{cc|cc} 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{array}\right].$$

Thus, every $2 \times 2 \times 2$ tensor can be transformed into one of the canonical forms listed in the statement of the theorem. Moreover, the invariants $\operatorname{sign}(\Delta)$ and $\operatorname{rank}_{\boxplus}$ are easily computed for the canonical forms and suffice to distinguish them. It follows that the listed forms are pairwise inequivalent.

We confirm the given values of $\operatorname{rank}_{\otimes}$. It is clear that $\operatorname{rank}_{\otimes}(D_0) = 0$ and $\operatorname{rank}_{\otimes}(D_1) = 1$. By Proposition 5.4, any tensor of rank 1 must be equivalent to D_1. Thus D_2, D_2', D_2'', and G_2 are all of rank 2. By Proposition 5.5, every tensor of rank 2 must be equivalent to one of these. In particular, D_3 and G_3 must have rank at least 3. Evidently $\operatorname{rank}_{\otimes}(D_3) = 3$ from its definition; and the same is true for G_3 by virtue of the less obvious relation

$$G_3 = (\mathbf{e}_1 + \mathbf{e}_2) \otimes \mathbf{e}_2 \otimes \mathbf{e}_2 + (\mathbf{e}_1 - \mathbf{e}_2) \otimes \mathbf{e}_1 \otimes \mathbf{e}_1 + \mathbf{e}_2 \otimes (\mathbf{e}_1 + \mathbf{e}_2) \otimes (\mathbf{e}_1 - \mathbf{e}_2).$$

Finally, we confirm the tabulated values of $\underline{\operatorname{rank}}_{\otimes}$. By virtue of the remark after Definition 5.11, it is enough to verify that $\underline{\operatorname{rank}}_{\otimes}(D_3) \leq 2$ and that $\underline{\operatorname{rank}}_{\otimes}(G_3) = 3$. The first of these assertions follows from Proposition 4.6. The set of tensors of type G_3 is an open set, which implies the second assertion. □

Remark. We note that D_3 is equivalent to any of the tensors obtained from it by permutations of the three factors. Indeed, all of these tensors have $\operatorname{rank}_{\boxplus} = (2, 2, 2)$ and $\Delta = 0$. Similar remarks apply to G_2, G_3.

Remark. The classification of $\mathrm{GL}_{2,2,2}(\mathbb{C})$-orbits in $\mathbb{C}^{2\times2\times2}$ differs only in the treatment of G_3, since there is no longer any distinction between real and complex eigenvalues.

We caution the reader that the *finite* classification in Theorem 7.1 is, in general, not possible for tensors of arbitrary size and order simply because the dimension or "degrees of freedom" of $\mathbb{R}^{d_1\times\cdots\times d_k}$ exceeds that of $\mathrm{GL}_{d_1,\ldots,d_k}(\mathbb{R})$ as soon as $d_1 \cdots d_k > d_1^2 + \cdots + d_k^2$ (which is almost always the case). Any attempt at an explicit classification must necessarily include continuous parameters. For the case of $\mathbb{R}^{2\times2\times2}$ this argument is not in conflict with our finite classification, since $2 \cdot 2 \cdot 2 < 2^2 + 2^2 + 2^2$.

7.1. Generic rank. We called the tensors in the orbit classes of G_2 and G_3 *generic* in the sense that the property of being in either one of these classes is an open condition. One should note that there is often no one single *generic outer-product rank* for tensors over $\mathbb{R}$ [50, 74]. (For tensors over $\mathbb{C}$ such a generic rank always exists [18].) The "generic outer-product rank" for tensors over $\mathbb{R}$ should be regarded as set-valued:

$$\text{generic-rank}_{\otimes}(\mathbb{R}^{d_1\times\cdots\times d_k}) = \{r \in \mathbb{N} \mid \mathcal{S}_r(d_1,\dots,d_k) \text{ has nonempty interior}\}.$$

So the generic outer-product rank in $\mathbb{R}^{2\times2\times2}$ is $\{2,3\}$. Another term, preferred by some and coined originally by ten Berge, is *typical rank* [74].

Given $d_1,\dots d_k$, the determination of the generic outer-product rank is an open problem in general and a nontrivial problem even in simple cases; see [13, 14] for results over $\mathbb{C}$ and [73, 74] for results over $\mathbb{R}$. Fortunately, the difficulty does not extend to multilinear rank; a single unique *generic multilinear rank* always exists and depends only on $d_1,\dots d_k$ (and not on the base field; cf. Proposition 7.4).

PROPOSITION 7.2. *Let* $A \in \mathbb{R}^{d_1\times\cdots\times d_k}$. *If* $\operatorname{rank}_{\boxplus}(A) = (r_1(A),\dots,r_k(A))$, *then*

$$r_i(A) = \min\left(d_i, \prod_{j\neq i} d_j\right), \quad i = 1,\dots,k,$$

generically.

Proof. Let $\mu_i : \mathbb{R}^{d_1\times\cdots\times d_k} \to \mathbb{R}^{d_i\times\prod_{j\neq i} d_j}$ be the forgetful map that "flattens" or "unfolds" a tensor into a matrix in the ith mode. It is easy to see that

$$r_i(A) = \operatorname{rank}(\mu_i(A)), \tag{7.1}$$

where "rank" here denotes matrix rank. The results then follow from the fact that the generic rank of a matrix in $\mathbb{R}^{d_i\times\prod_{j\neq i} d_j}$ is $\min(d_i, \prod_{j\neq i} d_j)$. ∎

For example, for order-3 tensors,

$$\text{generic-rank}_{\boxplus}(\mathbb{R}^{l\times m\times n}) = (\min(l, mn), \min(m, ln), \min(n, lm)).$$

7.2. Semialgebraic description of orbit classes. For a general tensor $A \in \mathbb{R}^{2\times2\times2}$, its orbit class is readily determined by computing the invariants $\operatorname{sign}(\Delta(A))$ and $\operatorname{rank}_{\boxplus}(A)$ and comparing with the canonical forms. The ranks $r_i(A)$ which constitute $\operatorname{rank}_{\boxplus}(A)$ can be evaluated algebraically as follows. If $A \neq 0$, then each $r_i(A)$ is either 1 or 2. For example, note that $r_1(A) < 2$ if and only if the vectors $A_{\bullet11}$, $A_{\bullet12}$, $A_{\bullet21}$, $A_{\bullet22}$ are linearly dependent, which happens if and only if all the 2-by-2 minors of the matrix

$$\begin{bmatrix} a_{111} & a_{112} & a_{121} & a_{122} \\ a_{211} & a_{212} & a_{221} & a_{222} \end{bmatrix}$$

are zero. Explicitly, the following six equations must be satisfied:

$$\begin{aligned} &a_{111}a_{212} = a_{211}a_{112}, \quad a_{111}a_{221} = a_{211}a_{121}, \quad a_{111}a_{222} = a_{211}a_{122}, \\ &a_{112}a_{221} = a_{212}a_{121}, \quad a_{112}a_{222} = a_{212}a_{122}, \quad a_{121}a_{222} = a_{221}a_{122}. \end{aligned} \tag{7.2}$$

Similarly, $r_2(A) < 2$ if and only if

$$\begin{aligned} &a_{111}a_{122} = a_{121}a_{112}, \quad a_{111}a_{221} = a_{121}a_{211}, \quad a_{111}a_{222} = a_{121}a_{212}, \\ &a_{112}a_{122} = a_{122}a_{211}, \quad a_{112}a_{222} = a_{122}a_{212}, \quad a_{211}a_{222} = a_{221}a_{212}; \end{aligned} \tag{7.3}$$

and $r_3(A) < 2$ if and only if

$$\begin{aligned} &a_{111}a_{122} = a_{112}a_{121}, \quad a_{111}a_{212} = a_{112}a_{211}, \quad a_{111}a_{222} = a_{112}a_{221}, \\ &a_{121}a_{212} = a_{122}a_{211}, \quad a_{121}a_{222} = a_{122}a_{221}, \quad a_{211}a_{222} = a_{212}a_{221}. \end{aligned} \tag{7.4}$$

The equations (7.2)–(7.4) lead to twelve distinct polynomials (beginning with $\Delta_1 = a_{111}a_{212} - a_{211}a_{112}$) which, together with $\Delta_0 := \Delta$, make up the collection $\Delta_0, \dots, \Delta_{12}$ of polynomials used in the semialgebraic description of the orbit structure of $\mathbb{R}^{2\times2\times2}$, as in Corollary 6.3. Indeed, we note that in Table 7.1 the information in the fourth and fifth columns ($\operatorname{rank}_\otimes(A)$, $\underline{\operatorname{rank}}_\otimes(A)$) is determined by the information in the second and third columns ($\operatorname{sign}(\Delta)$, $\operatorname{rank}_\boxplus(A)$).

7.3. Generic rank on $\Delta = 0$. The notion of generic rank also makes sense on subvarieties of $\mathbb{R}^{2\times2\times2}$—for instance, on the $\Delta = 0$ hypersurface.

PROPOSITION 7.3. *The tensors on the hypersurface* $\mathcal{D}_3 = \{A \in \mathbb{R}^{2\times2\times2} \mid \Delta(A) = 0\}$ *are all of the form*

$$\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{y}_3 + \mathbf{x}_1 \otimes \mathbf{y}_2 \otimes \mathbf{x}_3 + \mathbf{y}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3,$$

and they have rank 3 *generically.*

Proof. From the canonical forms in Table 7.1, we see that if $\Delta(A) = 0$, then

$$A = \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{y}_3 + \mathbf{x}_1 \otimes \mathbf{y}_2 \otimes \mathbf{x}_3 + \mathbf{y}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3$$

for some $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^2$, not necessarily linearly independent. It remains to be shown that $\operatorname{rank}_\otimes(A) = 3$ generically.

From Theorem 7.1 and the subsequent discussion, if $\Delta(A) = 0$, then $\operatorname{rank}_\otimes(A) \le 2$ if and only if at least one of the equation sets (7.2), (7.3), (7.4) is satisfied. Hence $\mathcal{D}_2 := \{A \mid \Delta(A) = 0, \operatorname{rank}_\otimes(A) \le 2\}$ is an algebraic subset of $\mathcal{D}_3$.

On the other hand, $\mathcal{D}_3 \setminus \mathcal{D}_2$ is dense in $\mathcal{D}_3$ with respect to the Euclidean, and hence the Zariski, topology. Indeed, each of the tensors D_0, D_1, D_2, D_2', D_2'' can be approximated by tensors of type D_3; for instance,

$$\left[\begin{array}{cc|cc} 1 & 0 & 0 & \epsilon \\ 0 & 1 & 0 & 0 \end{array}\right] \to \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right] = D_2 \quad \text{as } \epsilon \to 0.$$

Multiplying by an arbitrary $(L, M, N) \in \mathrm{GL}_{2,2,2}(\mathbb{R})$, it follows that any tensor in $\mathcal{D}_2$ can be approximated by tensors of type D_3.

It follows that the rank-3 tensors $\mathcal{D}_3 \setminus \mathcal{D}_2$ in $\mathcal{D}_3$ constitute a generic subset of $\mathcal{D}_3$ in the Zariski sense (and hence in all the other usual senses). ∎

Remark. In fact, it can be shown that $\mathcal{D}_3$ is an irreducible variety. If we accept that, then the fact that $\mathcal{D}_2$ is a proper subvariety of $\mathcal{D}_3$ immediately implies that the rank-3 tensors form a generic subset of $\mathcal{D}_3$. The denseness argument becomes unnecessary.

7.4. Base field dependence. It is interesting to observe that the $\mathrm{GL}_{2,2,2}(\mathbb{R})$-orbit classes of G_2 and G_3 merge into a single orbit class over $\mathbb{C}$ (under the action of $\mathrm{GL}_{2,2,2}(\mathbb{C})$). Explicitly, if we write $\mathbf{z}_k = \mathbf{x}_k + i\mathbf{y}_k$ and $\bar{\mathbf{z}}_k = \mathbf{x}_k - i\mathbf{y}_k$, then

$$\begin{aligned} &\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 + \mathbf{x}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3 - \mathbf{y}_1 \otimes \mathbf{x}_2 \otimes \mathbf{y}_3 + \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{x}_3 \\ &\qquad = \frac{1}{2}(\bar{\mathbf{z}}_1 \otimes \mathbf{z}_2 \otimes \bar{\mathbf{z}}_3 + \mathbf{z}_1 \otimes \bar{\mathbf{z}}_2 \otimes \mathbf{z}_3). \end{aligned} \tag{7.5}$$

The LHS is in the $\mathrm{GL}_{2,2,2}(\mathbb{R})$-orbit class of G_3 and has outer-product rank 3 over $\mathbb{R}$. The RHS is in the $\mathrm{GL}_{2,2,2}(\mathbb{C})$-orbit class of G_2 and has outer-product rank 2 over $\mathbb{C}$. To see why this is unexpected, recall that an $m \times n$ matrix with real entries has the same rank whether we regard it as an element of $\mathbb{R}^{m\times n}$ or of $\mathbb{C}^{m\times n}$. Note, however, that G_2 and G_3 have the same multilinear rank; this is not coincidental but is a manifestation of the following result.

PROPOSITION 7.4. *The multilinear rank of a tensor is independent of the choice of base field. If $\mathbb{K}$ is an extension field of $\Bbbk$, the value* $\operatorname{rank}_{\boxplus}(A)$ *is the same whether A is regarded as an element of $\Bbbk^{d_1\times\cdots\times d_k}$ or of $\mathbb{K}^{d_1\times\cdots\times d_k}$.*

Proof. This follows immediately from (7.1) and the base field independence of matrix rank. □

In 1969, Bergman [4] considered linear subspaces of matrix spaces, and showed that the minimum rank on a subspace can become strictly smaller upon taking a field extension. He gave a class of examples, the simplest instance being the 2-dimensional subspace

$$s\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} + t\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}$$

of $\mathbb{R}^{2\times 2}$. Every (nonzero) matrix in this subspace has rank 2, but the complexified subspace contains a matrix of rank 1. Intriguingly, this example is precisely the subspace spanned by the slabs of G_3. We suspect a deeper connection.

7.5. Injectivity of orbits. The tensor rank has the property of being invariant under the general multilinear group (cf. (2.15)). Indeed, much of its relevance comes from this fact. Moreover, from Proposition 3.1 we know that tensor rank is preserved when a tensor space is included in a larger tensor space. Similar assertions are true for the multilinear rank (cf. (2.19)).

The situation is more complicated for the function Δ defined on $\mathbb{R}^{2\times2\times2}$. The sign of Δ is $\mathrm{GL}_{2,2,2}(\mathbb{R})$-invariant, and Δ itself is invariant under $\mathrm{O}_{2,2,2}(\mathbb{R})$. For general $d_1, d_2, d_3 \geq 2$, we do not have an obvious candidate function Δ defined on $\mathbb{R}^{d_1\times d_2\times d_3}$. However, there is a natural definition of Δ restricted to the subset of tensors A for which $\operatorname{rank}_{\boxplus}(A) \leq (2,2,2)$. Such a tensor can be expressed as

$$A = (L, M, N)\cdot(B \oplus 0),$$

where $B \in \mathbb{R}^{2\times2\times2}$, $0 \in \mathbb{R}^{(d_1-2)\times(d_2-2)\times(d_3-2)}$, and $(L, M, N) \in \mathrm{O}_{d_1,d_2,d_3}(\mathbb{R})$. We provisionally define $\Delta(A) = \Delta(B)$, subject to a check that this is independent of the choices involved. Given an alternative expression $A = (L', M', N')\cdot(B' \oplus 0)$, it follows that $B \oplus 0$ and $B' \oplus 0$ are in the same $\mathrm{O}_{d_1,d_2,d_3}(\mathbb{R})$-orbit. Indeed,

$$B \oplus 0 = (L^{-1}L', M^{-1}M', N^{-1}N')\cdot(B' \oplus 0).$$

If we can show, more strongly, that B, B' belong to the same $\mathrm{O}_{2,2,2}(\mathbb{R})$-orbit, then the desired equality $\Delta(B) = \Delta(B')$ follows from the orthogonal invariance of Δ.

The missing step is supplied by the next theorem, which we state in a basis-free form for abstract vector spaces. If V is a vector space, we write $\mathrm{GL}(V)$ for the group of invertible linear maps from $V \to V$. If, in addition, V is an inner-product space, we write $\mathrm{O}(V)$ for the group of norm-preserving linear maps $V \to V$. In particular, $\mathrm{GL}(\mathbb{R}^d) \cong \mathrm{GL}_d(\mathbb{R})$ and $\mathrm{O}(\mathbb{R}^d) \cong \mathrm{O}_d(\mathbb{R})$.

THEOREM 7.5 (injectivity of orbits). *Let $\Bbbk = \mathbb{R}$ or $\mathbb{C}$ and $V_1, \ldots, V_k$ be $\Bbbk$-vector spaces. Let $U_1 \leq V_1, \ldots, U_k \leq V_k$.* (1) *Suppose $B, B' \in U_1 \otimes \cdots \otimes U_k$ are in*

distinct $\mathrm{GL}(U_1) \times \cdots \times \mathrm{GL}(U_k)$*-orbits of* $U_1 \otimes \cdots \otimes U_k$*; then* B *and* B' *are in distinct* $\mathrm{GL}(V_1) \times \cdots \times \mathrm{GL}(V_k)$*-orbits of* $V_1 \otimes \cdots \otimes V_k$*.* (2) *Suppose* $B, B' \in U_1 \otimes \cdots \otimes U_k$ *are in distinct* $\mathrm{O}(U_1) \times \cdots \times \mathrm{O}(U_k)$*-orbits of* $U_1 \otimes \cdots \otimes U_k$*; then* B *and* B' *are in distinct* $\mathrm{O}(V_1) \times \cdots \times \mathrm{O}(V_k)$*-orbits of* $V_1 \otimes \cdots \otimes V_k$*.*

LEMMA 7.6. *Let* $W \leq U \leq V$ *be vector spaces and* $L \in \mathrm{GL}(V)$*. Suppose* $L(W) \leq U$*. Then there exists* $\tilde{L} \in \mathrm{GL}(U)$ *such that* $L|_W = \tilde{L}|_W$*. Moreover, if* $L \in \mathrm{O}(V)$*, then we can take* $\tilde{L} \in \mathrm{O}(U)$*.*

Proof. Extend $L|_W$ to U by mapping the orthogonal complement of W in U by a norm-preserving map to the orthogonal complement of $L(W)$ in U. The resulting linear map $\tilde{L}$ has the desired properties and is orthogonal if L is orthogonal. ∎

Proof of Theorem 7.5. We prove the contrapositive form of the theorem. Suppose $B' = (L_1, \ldots, L_k) \cdot B$, where $L_i \in \mathrm{GL}(V_i)$. Let $W_i \leq U_i$ be minimal subspaces such that B is in the image of $W_1 \otimes \cdots \otimes W_k \hookrightarrow U_1 \otimes \cdots \otimes U_k$. It follows that $L_i(W_i) \leq U_i$, for otherwise we could replace W_i by $L_i^{-1}(L_i(W_i) \cap U_i)$. We can now use Lemma 7.6 to find $\tilde{L}_i \in \mathrm{GL}(U_i)$ which agree with L_i on W_i. By construction, $(\tilde{L}_1, \ldots, \tilde{L}_k) \cdot B = (L_1, \ldots, L_k) \cdot B = B'$. In the orthogonal case, where $L_i \in \mathrm{O}(V_i)$, we may choose $\tilde{L}_i \in \mathrm{O}(U_i)$. ∎

COROLLARY 7.7. *Let* φ *be a* $\mathrm{GL}_{d_1,\ldots,d_k}(\mathbb{R})$*-invariant (respectively,* $\mathrm{O}_{d_1,\ldots,d_k}(\mathbb{R})$*-invariant) function on* $\mathbb{R}^{d_1 \times \cdots \times d_k}$*. Then* φ *naturally extends to a* $\mathrm{GL}_{d_1,\ldots,d_k}(\mathbb{R})$*-invariant (respectively,* $\mathrm{O}_{d_1,\ldots,d_k}(\mathbb{R})$*-invariant) function on the subset*

$$\{A \in \mathbb{R}^{(d_1+e_1)\times\cdots\times(d_k+e_k)} \mid r_i(A) \leq d_i \text{ for } i = 1, \ldots, k\}$$

of $\mathbb{R}^{(d_1+e_1)\times\cdots\times(d_k+e_k)}$*.*

Proof. As with Δ above, write $A = (L_1, \ldots, L_k) \cdot B$ for $B \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ and define $\varphi(A) = \varphi(B)$. By Theorem 7.5 this is independent of the choices involved. ∎

The problem of classification is closely related to finding invariant functions. We end this section with a strengthening of Theorem 7.1.

COROLLARY 7.8. *The eight orbits in Theorem* 7.1 *remain distinct under the embedding* $\mathbb{R}^{2\times2\times2} \hookrightarrow \mathbb{R}^{d_1\times d_2\times d_3}$ *for any* $d_1, d_2, d_3 \geq 2$*. Thus, Theorem* 7.1 *immediately gives a classification of tensors* $A \in \mathbb{R}^{d_1\times d_2\times d_3}$ *with* $\operatorname{rank}_{\boxplus}(A) \leq (2, 2, 2)$*, into eight classes under* $\mathrm{GL}_{d_1,d_2,d_3}(\mathbb{R})$*-equivalence.*

The corollary allows us to extend the notion of tensor type to $\mathbb{R}^{d_1\times d_2\times d_3}$. For instance, we will say that $A \in \mathbb{R}^{d_1\times d_2\times d_3}$ has type G_3 if and only if A is GL-equivalent to $G_3 \in \mathbb{R}^{2\times2\times2} \subset \mathbb{R}^{d_1\times d_2\times d_3}$.

Note that order-k tensors can be embedded in order-$(k+1)$ tensors by taking the tensor product with a 1-dimensional factor. Distinct orbits remain distinct, so the results of this subsection extend to inclusions into tensor spaces of higher order.

8. Volume of tensors with no optimal low-rank approximation. At this point, it is clear that there exist tensors that can fail to have optimal low-rank approximations. However, it is our experience that practitioners have sometimes expressed optimism that such failures might be rare abnormalities that are not encountered in practice. In truth, such optimism is misplaced: the set of tensors with no optimal low-rank approximation has positive volume. In other words, a randomly chosen tensor will have a nonzero chance of failing to have a optimal low-rank approximation.

We begin this section with a particularly striking instance of this.

THEOREM 8.1. *No tensor of rank* 3 *in* $\mathbb{R}^{2\times2\times2}$ *has an optimal rank-*2 *approximation (with respect to the Frobenius norm). In particular,* $\textsc{approx}(A, 2)$ *has no solution for tensors of type* G_3*, which comprise a set that is open and therefore of positive volume.*

LEMMA 8.2. *Let* $A \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ *with* $\operatorname{rank}_\otimes(A) \geq r$. *Suppose* $B \in \mathcal{S}_r(d_1, \ldots, d_k)$ *is an optimal rank-r approximation for A. Then* $\operatorname{rank}_\otimes(B) = r$.

Proof. Suppose $\operatorname{rank}_\otimes(B) \leq r-1$. Then $B \neq A$, and so $B - A$ has at least one nonzero entry in its array representation. Let $E \in \mathbb{R}^{d_1 \times \cdots \times d_k}$ be the rank-1 tensor which agrees with $B - A$ at that entry and is zero everywhere else. Then $\operatorname{rank}_\otimes(B + E) \leq r$ but $\|A - (B + E)\|_F < \|A - B\|_F$, so B is not optimal. ∎

Proof of Theorem 8.1. Let $A \in \mathbb{R}^{2\times 2\times 2}$ have rank 3, and suppose that B is an optimal rank-2 approximation to A. Propositions 5.9 and 5.10, together with the continuity of Δ, imply that $\Delta(B) = 0$. Lemma 8.2 implies that $\operatorname{rank}_\otimes(B) = 2$. By Theorem 7.1, it follows that B is of type D_2, D_2', or D_2''.

We may assume without loss of generality that B is of type D_2. The next step is to put B into a helpful form by making an orthogonal change of coordinates. This gives an equivalent approximation problem, thanks to the O-invariance of the Frobenius norm. From Table 7.1, we know that $\operatorname{rank}_\boxplus(B) = (1, 2, 2)$. Such a B is orthogonally equivalent to a tensor of the following form:

$$\left[\begin{array}{cc|cc} \lambda & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \end{array}\right]. \tag{8.1}$$

Indeed, a rotation in the first tensor factor brings B entirely into the first slab, and further rotations in the second and third factors put the resulting matrix into diagonal form, with singular values $\lambda, \mu \neq 0$.

Henceforth we assume that B is equal to the tensor in (8.1). We will consider perturbations of the form $B + \epsilon H$, which will be chosen so that $\Delta(B + \epsilon H) = 0$ for all $\epsilon \in \mathbb{R}$. Then $B + \epsilon H \in \overline{\mathcal{S}}_2(2, 2, 2)$, and we must have

$$\|A - B\|_F \leq \|A - (B + \epsilon H)\|_F$$

for all ϵ. In fact

$$\|A - (B + \epsilon H)\|_F^2 - \|A - B\|_F^2 = -2\epsilon\langle A - B, H\rangle_F + \epsilon^2\|H\|_F^2,$$

so if this is to be nonnegative for all small values of ϵ, it is necessary that

$$\langle A - B, H\rangle_F = 0. \tag{8.2}$$

Tensors H which satisfy the condition $\Delta(B + \epsilon H) \equiv 0$ include the following:

$$\left[\begin{array}{cc|cc} \times & \times & 0 & 0 \\ \times & \times & 0 & 0 \end{array}\right], \quad \left[\begin{array}{cc|cc} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right], \quad \left[\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right], \quad \left[\begin{array}{cc|cc} 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \mu \end{array}\right]$$

since the resulting tensors have types D_2, D_3, D_3, and D_2, respectively.

Each of these gives a constraint on $A - B$, by virtue of (8.2). Putting the constraints together, we find that

$$A - B = \left[\begin{array}{cc|cc} 0 & 0 & a\mu & 0 \\ 0 & 0 & 0 & -a\lambda \end{array}\right] \qquad \text{or} \qquad A = \left[\begin{array}{cc|cc} \lambda & 0 & a\mu & 0 \\ 0 & \mu & 0 & -a\lambda \end{array}\right]$$

for some $a \in \mathbb{R}$. Thus $A = (\lambda\mathbf{e}_1 + a\mu\mathbf{e}_2) \otimes \mathbf{e}_1 \otimes \mathbf{e}_1 + (\mu\mathbf{e}_1 - a\lambda\mathbf{e}_2) \otimes \mathbf{e}_2 \otimes \mathbf{e}_2$ has rank 2, which is a contradiction. ∎

COROLLARY 8.3. *Let* $d_1, d_2, d_3 \geq 2$. *If* $A \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ *is of type* G_3, *then A does not have an optimal rank-2 approximation.*

Proof. We use the projection Π_A defined in subsection 2.6. For any $B \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, Pythagoras's theorem (2.20) gives

$$\begin{aligned}\|B - A\|_F^2 &= \|\Pi_A(B - A)\|_F^2 + \|(1 - \Pi_A)(B - A)\|_F^2 \\ &= \|\Pi_A(B) - A\|_F^2 + \|B - \Pi_A(B)\|_F^2.\end{aligned}$$

If B is an optimal rank-2 approximation, then it follows that $B = \Pi_A(B)$; for otherwise $\Pi_A(B)$ would be a better approximation. Thus $B \in U_1 \otimes U_2 \otimes U_3$, where U_1, U_2, U_3 are the supporting subspaces of A. These are 2-dimensional, since $\operatorname{rank}_{\boxplus}(A) = (2, 2, 2)$, so $U_1 \otimes U_2 \otimes U_3 \cong \mathbb{R}^{2\times 2\times 2}$. The optimality of B now contradicts Theorem 8.1. ∎

Our final result is that the set of tensors A for which $\textsc{approx}(A, 2)$ has no solution is a set of positive volume for all tensor spaces of order 3 except those isomorphic to a matrix space—in other words, Theorem 1.3. Note that the G_3-tensors comprise a set of zero volume in all cases except $\mathbb{R}^{2\times 2\times 2}$. Here is the precise statement.

THEOREM 8.4. *Let $d_1, d_2, d_3 \geq 2$. The set of tensors $A \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ for which* $\textsc{approx}(A, 2)$ *does not have a solution (in the Frobenius norm) contains an open neighborhood of the set of tensors of type G_3. In particular, this set is nonempty and has positive volume.*

For $A \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, let $\mathcal{B}(A)$ denote the set of optimal border-rank-2 approximations for A. Since $\overline{\mathcal{S}}_2(d_1, d_2, d_3)$ is nonempty and closed, it follows that $\mathcal{B}(A)$ is nonempty and compact.

We can restate the theorem as follows. Let A_0 be an arbitrary G_3-tensor. We must show that if A is close to A_0, and $B \in \mathcal{B}(A)$, then $\operatorname{rank}_{\otimes}(B) > 2$, i.e., B is a D_3-tensor. Our proof strategy is contained in the steps of the following lemma.

LEMMA 8.5. *Let $A_0 \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ be a fixed tensor of type G_3. Then there exist positive numbers $\rho = \rho(A_0)$, $\delta = \delta(A_0)$ such that the following statements are true for all $A \in \mathbb{R}^{d_1 \times d_2 \times d_3}$.*

1. *If A is a G_3-tensor and $B \in \mathcal{B}(A)$, then B is a D_3-tensor and $\Pi_B = \Pi_A$.*
2. *If $\|A - A_0\|_F < \rho$ and $\operatorname{rank}_{\boxplus}(A) \leq (2, 2, 2)$, then A is a G_3-tensor.*
3. *If $\|A - A_0\|_F < \delta$ and $B \in \mathcal{B}(A)$, define $A' = \Pi_B(A)$. Then $\|A' - A_0\|_F < \rho$ and $B \in \mathcal{B}(A')$.*

Proof of Theorem 8.4*, assuming Lemma* 8.5. Fix $A_0 \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ and suppose $\|A - A_0\|_F < \delta$. It is not generally true that $\operatorname{rank}_{\boxplus}(A) \leq (2, 2, 2)$, so we cannot apply (2) directly to A. Let $B \in \mathcal{B}(A)$. Then $A' = \Pi_B(A)$ is close to A_0, by (3). Since $\operatorname{rank}_{\boxplus}(B) \leq (2, 2, 2)$ and Π_B is the projection onto the subspace spanned by B, it follows that $\operatorname{rank}_{\boxplus}(A') \leq (2, 2, 2)$. Now (2) implies that A' is a G_3-tensor. Since $B \in \mathcal{B}(A')$, by (3), it follows from (1) that B is a D_3-tensor. ∎

Proof of Lemma 8.5 (1). This is essentially Corollary 8.3: B cannot have rank 2 or less, but it has border-rank 2, so B must be a D_3-tensor. Since $B = \Pi_A(B)$ it follows that the supporting subspaces of B are contained in the supporting subspaces of A. However, $\operatorname{rank}_{\boxplus}(B) = (2, 2, 2) = \operatorname{rank}_{\boxplus}(A)$, so the two tensors must have the same supporting subspaces, and so $\Pi_B = \Pi_A$. ∎

Proof of Lemma 8.5 (2). Let $\overline{\mathcal{S}}_2^+(d_1, d_2, d_3)$ denote the set of nonG_3 tensors in $\mathbb{R}^{d_1 \times d_2 \times d_3}$ with $\operatorname{rank}_{\boxplus} \leq (2, 2, 2)$. Since $A_0 \notin \overline{\mathcal{S}}_2^+(d_1, d_2, d_3)$, it is enough to show that $\overline{\mathcal{S}}_2^+(d_1, d_2, d_3)$ is closed, for then it would be disjoint from the ρ-ball about A_0 for some $\rho > 0$. Note that

$$\overline{\mathcal{S}}_2^+(d_1, d_2, d_3) = \mathrm{O}_{d_1,d_2,d_3}(\mathbb{R}) \cdot \overline{\mathcal{S}}_2^+(2, 2, 2).$$

Now $\overline{\mathcal{S}}_2^+(2, 2, 2) = \{A \in \mathbb{R}^{2\times 2\times 2} \mid \Delta(A) \geq 0\}$ is a closed subset of $\mathbb{R}^{2\times 2\times 2}$, and the

action of the compact group $\mathrm{O}_{d_1,d_2,d_3}(\mathbb{R})$ is proper. It follows that $\overline{\mathcal{S}}_2^+(d_1,d_2,d_3)$ is closed, as required. ∎

Proof of Lemma 8.5 (3). We begin with the easier part of the statement, which is that $B \in \mathcal{B}(A')$. To prove this, we will show that $\|A'-B\|_F \le \|A'-B'\|_F$ whenever $B' \in \mathcal{B}(A')$, establishing the optimality of B as an approximation to A'. Accordingly, let $B' \in \mathcal{B}(A')$. Since $\Pi_B(A') = A'$, it follows from (2.20) with Π_B that

$$\|A'-B'\|_F^2 = \|A'-\Pi_B(B')\|_F^2 + \|B'-\Pi_B(B')\|_F^2,$$

so, since B' is optimal, we must have $\Pi_B(B') = B'$. We can now apply (2.20) with Π_B to both sides of the inequality $\|A-B\|_F^2 \le \|A-B'\|_F^2$ to get

$$\|A'-B\|_F^2 + \|A-A'\|_F^2 \le \|A'-B'\|_F^2 + \|A-A'\|_F^2,$$

and hence $\|A'-B\|_F \le \|A'-B'\|_F$, as claimed.

We now turn to the proof that $\Pi_B(A)$ is close to A_0 if A is close to A_0. This is required to be uniform in A and B. In other words, there exists $\delta = \delta(A_0) > 0$ such that for all A and all $B \in \mathcal{B}(A)$ if $\|A-A_0\|_F < \delta$, then $\|\Pi_B(A) - A_0\| < \rho$. Here $\rho = \rho(A_0)$ is fixed from part (2) of this lemma.

We need control over the location of B. Let $\mathcal{B}_\epsilon(A_0)$ denote the ϵ-neighborhood of $\mathcal{B}(A_0)$ in $\overline{\mathcal{S}}_2(d_1,d_2,d_3)$.

Proposition 8.6. *Given $\epsilon > 0$, there exists $\delta > 0$ such that if $\|A-A_0\|_F < \delta$ then $\mathcal{B}(A) \subset \mathcal{B}_\epsilon(A_0)$.*

Proof. The set $\overline{\mathcal{S}}_2(d_1,d_2,d_3) \setminus \mathcal{B}_\epsilon(A_0)$ is closed, and so it attains its minimum distance from A_0. This must exceed the absolute minimum $\|A_0-B_0\|_F$ for $B_0 \in \mathcal{B}(A_0)$ by a positive quantity 2δ, say. If $\|A-A_0\|_F < \delta$ and $B' \in \overline{\mathcal{S}}_2(d_1,d_2,d_3) \setminus \mathcal{B}_\epsilon(A_0)$ then

$$\begin{aligned}
\|A-B'\|_F &\ge \|B'-A_0\|_F - \|A-A_0\|_F \\
&> \|A_0-B_0\|_F + 2\delta - \delta \\
&= \|A_0-B_0\|_F + \delta \\
&> \|A_0-B_0\|_F + \|A-A_0\|_F \\
&\ge \|A-B_0\|_F
\end{aligned}$$

using the triangle inequality in the first and last lines. Thus $B' \notin \mathcal{B}(A)$. ∎

We claim that if ϵ is small enough, then $\mathrm{rank}_{\boxplus}(B) = (2,2,2)$ for all $B \in \mathcal{B}_\epsilon(A_0)$. Indeed, this is already true on $\mathcal{B}(A_0)$, by part (1). Since $\mathrm{rank}_{\boxplus}$ is upper-semicontinuous and does not exceed $(2,2,2)$ on $\overline{\mathcal{S}}_2(d_1,d_2,d_3)$, it must be constant on a neighborhood of $\mathcal{B}(A_0)$ in $\overline{\mathcal{S}}_2(d_1,d_2,d_3)$. Since $\mathcal{B}(A_0)$ is compact, the neighborhood can be taken to be an ϵ-neighborhood.

Part (1) implies that $\Pi_{B_0} = \Pi_{A_0}$ for all $B_0 \in \mathcal{B}(A_0)$. If ϵ is small enough that $\mathrm{rank}_{\boxplus}(B) = (2,2,2)$ on $\mathcal{B}_\epsilon(A_0)$, then Π_B depends continuously on $B \in \mathcal{B}_\epsilon(A_0)$, by Proposition 2.5. Since $\mathcal{B}(A_0)$ is compact, we can choose ϵ small enough so that the operator norm of $\Pi_B - \Pi_{A_0}$ is as small as we like, uniformly over $\mathcal{B}_\epsilon(A_0)$.

We are now ready to confine $\Pi_B(A)$ to the ρ-neighborhood of A_0. Suppose, initially, that $\|A-A_0\|_F \le \rho/2$ and $B \in \mathcal{B}_\epsilon(A_0)$. Then

$$\begin{aligned}
\|\Pi_B(A) - A_0\|_F &\le \|(\Pi_B - \Pi_{A_0})\cdot A\|_F + \|\Pi_{A_0}\cdot A - A_0\|_F \\
&\le \|\Pi_B - \Pi_{A_0}\|\|A\|_F + \|\Pi_{A_0}\cdot(A-A_0)\|_F \\
&\le \|\Pi_B - \Pi_{A_0}\|(\|A_0\|_F + \rho/2) + \|A-A_0\|_F \\
&\le \|\Pi_B - \Pi_{A_0}\|(\|A_0\|_F + \rho/2) + \rho/2.
\end{aligned}$$

Now choose $\epsilon > 0$ so that the operator norm $\|\Pi_B - \Pi_{A_0}\|$ is kept small enough to guarantee that the RHS is less than ρ. For this ϵ, choose δ as given by Proposition 8.6. Ensure also that $\delta < \rho/2$.

Then, if $\|A - A_0\|_F < \delta$ and $B \in \mathcal{B}(A)$, we have $B \in \mathcal{B}_\epsilon(A_0)$. By the preceding calculation, $\|A' - A_0\|_F < \rho$. This completes the proof. □

9. Closing remarks. We refer interested readers to [17, 18, 57, 58] for a discussion of similar issues for symmetric tensors and nonnegative tensors. In particular, the reader will find in [18] an example of a symmetric tensor of symmetric rank r (r may be chosen to be arbitrarily large) that does not have a best symmetric-rank-2 approximation. In [57, 58], we show that such failures do not occur in the context of nonnegative tensors; a nonnegative tensor of nonnegative-rank r will always have a best nonnegative-rank-s approximation for any $s \leq r$.

In this paper we have focused our attention on the real case; the complex case has been studied in great detail in algebraic computational complexity theory and algebraic geometry. For the interested reader, we note that the rank-jumping phenomenon still occurs: Proposition 4.6 and its proof carry straight through to the complex case. On the other hand, there is no distinction between G_3- and G_2-tensors over the complex numbers; if $\Delta(A) \neq 0$, then A has rank 2. The results of section 8 have no direct analogue.

The major open question in tensor approximation is how to overcome the ill-posedness of APPROX(A, r). In general this will conceivably require an equivalent of Theorem 5.1 that characterizes the limit points of rank-r order-k tensors. It is our hope that some of the tools developed in our study, such as Theorems 5.2 and 7.5 (both of which apply to general r and k), may be used in future studies. The type of characterization in Corollary 5.12, for $r = 2$ and $k = 3$, is an example of what one might hope to achieve.

Acknowledgments. We thank the anonymous reviewers for some exceptionally helpful comments. We also gratefully acknowledge Joseph Landsberg and Bernd Sturmfels for enlightening pointers that helped improved sections 4.7 and 6. Lek-Heng Lim thanks Gene Golub for his encouragement and helpful discussions. Both authors thank Gunnar Carlsson and the Department of Mathematics, Stanford University, where some of this work was done.

REFERENCES

[1] W. A. Adkins and S. H. Weintraub, *Algebra: An Approach via Module Theory*, Grad. Texts in Math. 136, Springer-Verlag, New York, 1992.
[2] M. D. Atkinson and S. Lloyd, *Bounds on the ranks of some* 3*-tensors*, Linear Algebra Appl., 31 (1980), pp. 19–31.
[3] M. D. Atkinson and N. M. Stephens, *On the maximal multiplicative complexity of a family of bilinear forms*, Linear Algebra Appl., 27 (1979), pp. 1–8.
[4] G. M. Bergman, *Ranks of tensors and change of base field*, J. Algebra, 11 (1969), pp. 613–621.
[5] D. Bini, *Border rank of* $m \times n \times (mn - q)$ *tensors*, Linear Algebra Appl., 79 (1986), pp. 45–51.
[6] D. Bini, *Border rank of a* $p \times q \times 2$ *tensor and the optimal approximation of a pair of bilinear forms*, in Automata, Languages, and Programming, J. W. de Bakker and J. van Leeuwen eds., Lecture Notes in Comput. Sci. 85, Springer-Verlag, New York, 1980, pp. 98–108.
[7] D. Bini, M. Capovani, G. Lotti, and F. Romani, $O(n^{2.7799})$ *complexity for* $n \times n$ *approximate matrix multiplication*, Inform. Process. Lett., 8 (1979), pp. 234–235.
[8] D. Bini, G. Lotti, and F. Romani, *Approximate solutions for the bilinear form computational problem*, SIAM J. Comput., 9 (1980), pp. 692–697.
[9] N. Bourbaki, *Algebra* I: *Chapters* 1–3, Elem. Math., Springer-Verlag, Berlin, 1998.

[10] L. Brègman, *A relaxation method of finding a common point of convex sets and its application to the solution of problems in convex programming*, U.S.S.R. Comput. Math. and Math. Phys., 7 (1967), pp. 620–631.
[11] R. Bro, *Multi-Way Analysis in the Food Industry: Models, Algorithms, and Applications*, Ph.D. thesis, Universiteit van Amsterdam, Amsterdam, The Netherlands, 1998.
[12] P. Bürgisser, M. Clausen, and M.A. Shokrollahi, *Algebraic Complexity Theory*, Grundlehren Math. Wiss. 315, Springer-Verlag, Berlin, 1996.
[13] M. V. Catalisano, A. V. Geramita, and A. Gimigliano, *Higher secant varieties of the Segre varieties* $\mathbb{P}^1 \times \cdots \times \mathbb{P}^1$, J. Pure Appl. Algebra, 201 (2005), pp. 367–380.
[14] M. V. Catalisano, A. V. Geramita, and A. Gimigliano, *Ranks of tensors, secant varieties of Segre varieties and fat points*, Linear Algebra Appl., 355 (2002), pp. 263–285.
[15] A. Cayley, *On the theory of linear transformation*, Cambridge Math. J., 4 (1845), pp. 193–209.
[16] P. Comon, *Tensor decompostions: State of the art and applications*, in Mathematics in Signal Processing V (Coventry, UK, 2000), Inst. Math. Appl. Conf. Ser. 71, Oxford University Press, Oxford, UK, 2002, pp. 1–24.
[17] P. Comon, G. H. Golub, L.-H. Lim, and B. Mourrain, *Genericity and rank deficiency of high order symmetric tensors*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '06), Vol. 31, IEEE, Washington, D.C., 2006, pp. 125–128.
[18] P. Comon, G. H. Golub, L.-H. Lim, and B. Mourrain, *Symmetric tensors and symmetric tensor rank*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1254–1279.
[19] M. Coste, *An Introduction to Semialgebraic Geometry*, preprint, 2002. Available online from http://perso.univ-rennes1.fr/michel.coste/polyens/SAG.pdf.
[20] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.
[21] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-*1 *and rank-*$(R_1, \ldots, R_N)$ *approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.
[22] J. W. Demmel, *On condition numbers and the distance to the nearest ill-posed problem*, Numer. Math., 51 (1987), pp. 251–289.
[23] J. W. Demmel, *The geometry of ill-conditioning*, J. Complexity, 3 (1987), pp. 201–229.
[24] J. W. Demmel, *The probability that a numerical analysis problem is difficult*, Math. Comp., 50 (1988), pp. 449–480.
[25] A. Defant and K. Floret, *Tensor Norms and Operator Ideals*, North-Holland Math. Stud. 176, North-Holland, Amsterdam, 1993.
[26] I. S. Dhillon and J. A. Tropp, *Matrix nearness problems with Brègman divergences*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1120–1146.
[27] D. S. Dummit and R. M. Foote, *Abstract Algebra*, 3rd ed., John Wiley & Son, Hoboken, NJ, 2003.
[28] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.
[29] L.C. Evans, *Partial Differential Equations*, Grad. Stud. Math. 19, AMS, Providence, RI, 1998.
[30] I. M. Gelfand, M. M. Kapranov, and A. V. Zelevinsky, *Discriminants, Resultants, and Multidimensional Determinants*, Birkhäuser Boston, Boston, MA, 1994.
[31] I. M. Gelfand, M. M. Kapranov, and A. V. Zelevinsky, *Hyperdeterminants*, Adv. Math., 96 (1992), pp. 226–263.
[32] R. Geroch, *Mathematical Physics*, Chicago Lectures Phys., University of Chicago Press, Chicago, IL, 1985.
[33] G. H. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[34] W. Greub, *Multilinear Algebra*, 2nd ed., Springer-Verlag, New York, 1978.
[35] D. Grier, P. Huggins, B. Sturmfels, and J. Yu, *The Hyperdeterminant and Triangulations of the* 4*-Cube*, preprint, 2006. Available online from http://bio.math.berkeley.edu/4cube/index.html.
[36] A. Grothendieck, *Résumé de la théorie métrique des produits tensoriels topologiques*, Bol. Soc. Mat. São Paulo, 8 (1953), pp. 1–79.
[37] J. Harris, *Algebraic Geometry: A First Course*, Grad. Texts in Math. 133, Springer-Verlag, New York, 1998.
[38] R. A. Harshman, *Foundations of the* PARAFAC *Procedure: Models and Conditions for an Explanatory Multi-Modal Factor Analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[39] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp. 164–189.
[40] F. L. Hitchcock, *Multiple invariants and generalized rank of a p-way matrix or tensor*, J. Math. Phys., 7 (1927), pp. 39–79.
[41] T. W. Hungerford, *Algebra*, Grad. Texts in Math. 73, Springer-Verlag, New York, 1980.
[42] J. JáJá, *An addendum to Kronecker's theory of pencils*, SIAM J. Appl. Math., 37 (1979), pp. 700–712.
[43] J. JáJá and J. Takche, *On the validity of the direct sum conjecture*, SIAM J. Comput., 15 (1986), pp. 1004–1020.
[44] E. Kofidis and P. A. Regalia, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 863–884.
[45] E. Kofidis and P. A. Regalia, *Tensor approximation and signal processing applications*, in Structured Matrices in Mathematics, Computer Science, and Engineering I, Contemp. Math. 280, AMS, Providence, RI, 2001, pp. 103–133.
[46] T. G. Kolda, *Orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 243–255.
[47] T. G. Kolda, *A counterexample to the possibility of an extension of the Eckart–Young low-rank approximation theorem for the orthogonal rank tensor decomposition*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 762–767.
[48] D. E. Knuth, *The Art of Computer Programming* 2: *Seminumerical Algorithms*, 3rd ed., Addison–Wesley, Reading, MA, 1998.
[49] W. P. Krijnen, T. K. Dijkstra, and A. Stegeman, *On the Non-Existence of Optimal Solutions and the Occurrence of "Degeneracy" in the* candecomp/parafac *Model*, preprint, 2007.
[50] J. B. Kruskal, *Rank, decomposition, and uniqueness for* 3*-way and N-way arrays*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., Elsevier Science, Amsterdam, 1989, pp. 7–18.
[51] J. B. Kruskal, R. A. Harshman, and M. E. Lundy, *How* 3*-MFA data can cause degenerate* parafac *solutions, among other relationships*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., Elsevier Science, Amsterdam, 1989, pp. 115–122.
[52] S. Lang, *Algebra*, rev. 3rd ed., Grad. Texts in Math. 211, Springer-Verlag, New York, 2002.
[53] J. M. Landsberg, *Private e-mail communication with the authors*, August 29, 2006.
[54] J. M. Landsberg, *The border rank of the multiplication of* 2×2 *matrices is seven*, J. Amer. Math. Soc., 19 (2006), pp. 447–459.
[55] J. M. Landsberg and L. Manivel, *On the ideals of secant varieties of Segre varieties*, Found. Comput. Math., 4 (2004), pp. 397–422.
[56] D. Leibovici and R. Sabatier, *A singular value decomposition of a k-way array for a principal component analysis of multiway data, PTA-k*, Linear Algebra Appl., 269 (1998), pp. 307–329.
[57] L.-H. Lim, *Optimal Solutions to Non-Negative* parafac*/multilinear* nmf *Always Exist*, Centre International de rencontres Mathématiques, Luminy, France, 2005.
[58] L.-H. Lim and G. H. Golub, *Nonnegative Decompositions of Nonnegative Matrices and Tensors*, SCCM tech. rep. 06-01, preprint, Stanford University, Stanford, CA, 2006.
[59] M. Marcus, *Finite Dimensional Multilinear Algebra, Part* I, Pure and Applied Mathematics 23, Marcel Dekker, New York, 1973.
[60] M. Marcus, *Finite Dimensional Multilinear Algebra, Part* II, Pure and Applied Mathematics 23, Marcel Dekker, New York, 1975.
[61] D. Martin, *Manifold Theory: An Introduction for Mathematical Physicists*, Ellis Horwood, NY, 1991.
[62] D. G. Northcott, *Multilinear Algebra*, Cambridge University Press, Cambridge, UK, 1984.
[63] P. Paatero, *Construction and analysis of degenerate* parafac *models*, J. Chemometrics, 14 (2000), pp. 285–299.
[64] J. J. Rotman, *Advanced Modern Algebra*, Prentice–Hall, Upper Saddle River, NJ, 2002.
[65] A. Seidenberg, *A new decision method for elementary algebra*, Ann. of Math., 60 (1954), pp. 365–374.
[66] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, *Parallel factor analysis in sensor array processing*, IEEE Trans. Signal Process., 48 (2000), pp. 2377–2388.
[67] A. Smilde, R. Bro, and P. Geladi, *Multi-Way Analysis: Applications in the Chemical Sciences*, John Wiley, West Sussex, UK, 2004.
[68] A. Stegeman, *Degeneracy in* candecomp/parafac *explained for* $p \times p \times 2$ *arrays of rank* $p+1$ *or higher*, Psychometrika, 71 (2006), pp. 483–501.
[69] A. Stegeman, *Low-rank approximation of generic* $p \times q \times 2$ *arrays and diverging components in the* candecomp/parafac *model*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 988–1007.

[70] V. Strassen, *Vermeidung von Divisionen*, J. Reine Angew. Math., 264 (1973), pp. 184–202.
[71] B. Sturmfels, *Private e-mail communication with the authors*, December 4, 2005.
[72] A. Tarski, *A Decision Method for Elementary Algebra and Geometry*, 2nd ed., University of California Press, Berkeley, CA, 1951.
[73] J. M. F. ten Berge, *Kruskal's polynomial for* $2 \times 2 \times 2$ *arrays and a generalization to* $2 \times n \times n$ *arrays*, Psychometrika, 56 (1991), pp. 631–636.
[74] J. M. F. ten Berge and H. A. L. Kiers, *Simplicity of core arrays in three-way principal component analysis and the typical rank of* $p \times q \times 2$ *arrays*, Linear Algebra Appl., 294 (1999), pp. 169–179.
[75] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear image analysis for facial recognition*, Proceedings of the International Conference on Pattern Recognition (ICPR), Quebec, Canada, 2002, pp. 511–514.
[76] M. A. O. Vasilescu and D. Terzopoulos, *TensorTextures: Multilinear image-based rendering*, in Proceedings of the ACM SIGGRAPH, ACM, New York, 2004, pp. 336–342.
[77] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice–Hall, Englewood Cliffs, NJ, 1963.
[78] T. Yokonuma, *Tensor Spaces and Exterior Algebra*, Trans. Math. Monogr. 108, AMS, Providence, RI, 1992.
[79] F. L. Zak, *Tangents and Secants of Algebraic Varieties*, Trans. Math. Monogr. 127, AMS, Providence, RI, 1993.
[80] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1128–1147

ENHANCED LINE SEARCH: A NOVEL METHOD TO ACCELERATE PARAFAC*

MYRIAM RAJIH†, PIERRE COMON†, AND RICHARD A. HARSHMAN‡

Abstract. Several modifications have been proposed to speed up the alternating least squares (ALS) method of fitting the PARAFAC model. The most widely used is line search, which extrapolates from linear trends in the parameter changes over prior iterations to estimate the parameter values that would be obtained after many additional ALS iterations. We propose some extensions of this approach that incorporate a more sophisticated extrapolation, using information on nonlinear trends in the parameters and changing all the parameter sets simultaneously. The new method, called "enhanced line search (ELS)," can be implemented at different levels of complexity, depending on how many different extrapolation parameters (for different modes) are jointly optimized during each iteration. We report some tests of the simplest parameter version, using simulated data. The performance of this lowest-level of ELS depends on the nature of the convergence difficulty. It significantly outperforms standard LS when there is a "convergence bottleneck," a situation where some modes have almost collinear factors but others do not, but is somewhat less effective in classic "swamp" situations where factors are highly collinear in all modes. This is illustrated by examples. To demonstrate how ELS can be adapted to different N-way decompositions, we also apply it to a four-way array to perform a blind identification of an under-determined mixture (UDM). Since analysis of this dataset happens to involve a serious convergence "bottleneck" (collinear factors in two of the four modes), it provides another example of a situation in which ELS dramatically outperforms standard line search.

Key words. PARAFAC, alternating least squares (ALS), line search, enhanced line search (ELS), acceleration, swamps, bottlenecks, collinear factors, degeneracy

AMS subject classifications. 65B99, 15A69, 15A21

DOI. 10.1137/06065577

1. Introduction.

PARAFAC can be seen as a generalization of two-way factor analysis to multiway data. It was first introduced by Harshman in 1970 [9] based on the principle of parallel proportional profiles (PP) proposed by Cattell in 1944 [4]. The PP principle states that if two (or more) different two-way models are described by the same set of loading vectors but their relative proportions or weights change from one model to the other, then those loading vectors lead to a new model which is unambiguous with respect to (w.r.t.) rotation [4, 5, 2]. In other words, suppose that the matrix $\mathbf{X}_1$ can be modeled:

$$\mathbf{X}_1 = \mathbf{a}_1\mathbf{b}_1^T c_{11} + \mathbf{a}_2\mathbf{b}_2^T c_{12} + \cdots + \mathbf{a}_F\mathbf{b}_F^T c_{1F} + \mathbf{E}_1,$$

where $\mathbf{a}_f$ and $\mathbf{b}_f$ $(1 \le f \le F)$ are the columns of matrices $\mathbf{A}$ and $\mathbf{B}$, respectively, and $\mathbf{E}_1$ is a matrix of random disturbances (and/or other unmodeled variation). And suppose that another matrix $\mathbf{X}_2$ can be modeled using the same set of loading vectors only in different proportions (i.e., $\frac{c_{11}}{c_{21}} \neq \frac{c_{12}}{c_{22}} \neq \cdots \frac{c_{1F}}{c_{2F}}$):

$$\mathbf{X}_2 = \mathbf{a}_1\mathbf{b}_1^T c_{21} + \mathbf{a}_2\mathbf{b}_2^T c_{22} + \cdots + \mathbf{a}_F\mathbf{b}_F^T c_{2F} + \mathbf{E}_2.$$

*Received by the editors March 29, 2006; accepted for publication (in revised form) by L. De Lathauwer July 5, 2007; published electronically September 25, 2008. This work has been partially supported by the IST Programme of the European Community, under the PASCAL Network of Excellence IST-2002-506778, and by the contract ANR-06-BLAN-0074 "DECOTES."
http://www.siam.org/journals/simax/30-3/65537.html

†I3S Laboratory, UNSA-CNRS, 2000 route des Lucioles, B.P. 121, F-06903 Sophia-Antipolis, France (rajih@i3s.unice.fr, comon@i3s.unice.fr).

‡Department of Psychology, University of Western Ontario, London, Ontario, N6A 5C2 Canada (harshman@uwo.ca, http://publish.uwo.ca/∼harshman).

Then, we can build a combined model:

$$\mathbf{X}_k = \mathbf{A}\mathbf{C}_k\mathbf{B}^T + \mathbf{E}_k, k = 1, 2, \tag{1}$$

where $\mathbf{C}_k$ is a diagonal matrix with the elements of vector $\mathbf{c}_k$ in its diagonal, where $\mathbf{c}_k$ denotes the kth row of the slice weighting or "occasion weights" matrix $\mathbf{C}$ [13]. The trilinear decomposition used in the model is also known as CANDECOMP for CANonical DECOMPosition; it was introduced by Caroll and Chang in 1970 [3] to provide a basis for fitting INDSCAL, an important generalization of multidimensional scaling that provides unique dimensions and allows the estimation of dimension weights for individual subjects. Alternatively, the model can be written in scalar form as

$$X_{ijk} = \sum_f A_{if}B_{jf}C_{kf} + E_{ijk}.$$

Matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are called loading matrices.

The three-way PARAFAC model, along with its extension to higher orders [9, 3], has most often been applied in psychometrics and chemometrics [26, 27], and in the signal processing area [18, 6, 7]. While the two-way factor model suffers a rotational indeterminacy that yields an infinite set of solutions, the PARAFAC model enjoys a uniqueness property under conditions that often can be met in real data situations. Uniqueness properties have been studied by multiple authors, with some of the most important general results found in [10] and its recent generalization [15], the Kruskal theorems in, e.g., [16], and the extensions in [26], and elsewhere. Progress in this area is ongoing—for example, in the case of "tall" arrays, a significantly more relaxed condition has been derived in [19]. A relatively complete list of relevant articles up to 2006-07 can be found in [15].

A variety of algorithms have been used to fit the PARAFAC model (for a detailed summary and discussion; see, e.g., [28]. The most widely used is the alternating least squares (ALS) algorithm. The convergence of ALS was found to be very slow in some cases, typically when two factors are almost collinear. Line search [2, 24] is one of the most important solutions proposed to cope with the problem of slow convergence. We focus in this paper on the line search solution and present a generalization of this method for speeding up ALS; we discuss its simplest version and demonstrate that it can exhibit very good performance in some circumstances, yet perform less successfully in others—opening interesting directions for further exploration. We call this method enhanced line search (ELS).

A regularized (ridge) regression was proposed by Rayens and Mitchell in [23] to speed up the ALS algorithm in case of ill-posed problems. While the estimates produced by ridge regression are biased, they suggested ways of dealing with this, including a switch back to regular ALS estimation at the end of the fitting procedure, when the approximate solution has been reached. They designed their method to avoid difficulties which they called convergence "swamps," characterized by high factor collinearity in all three modes. We will see that ELS (at least the simple version tested here) is most successful with a different kind of convergence difficulty.

In [22], Paatero proposed the multilinear engine (ME) program to accelerate the fit of the PARAFAC model. ME changes all of the sets of parameters at once, whereas ELS is based on ALS, and updates alternatively each of the loading factors.

In [8], Franc proposed an acceleration to the convergence of PARAFAC based on a gradient method. In fact, the loading matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are updated using the gradient descent.

A closed-form solution to fit the PARAFAC model was proposed by Sanchez and Kowalski in [25]. It reduces the problem to a rectangular eigenvalue-eigenvector equation, but it needs at least two of the loading matrices to be linearly independent.

Another closed-form solution for three-way arrays, and based on a single matrix eigenvalue decomposition (EVD) was presented in [21] by Leurgans, Ross, and Abel. It also requires that two of the loading matrices are linearly independent and that every pair of columns of the last loading matrix is linearly independent. Previous approaches are made more robust in [20] by taking all matrix slices into account, which leads to a simultaneous matrix decomposition. All of these methods require that the array rank (as defined in [17], for instance), F, is less than or equal to two of the array dimensions. In [19] De Lathauwer generalizes the approach presented in [20] to the case where F is less than or equal to one of the array's dimensions, and subject to a condition involving the product of the remaining array dimensions. One advantage of ELS is that it can be applied even if the previous conditions are not met.

2. Model and notation. We consider the three-way PARAFAC model of expression (1). This model can be written in a compact form using the Khatri–Rao product $\odot$ (columnwise Kronecker product) as, possibly up to an error term,

$$\mathbf{X}^{(I\times JK)} \approx \mathbf{A}(\mathbf{C}\odot\mathbf{B})^T,$$

where matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are matrices of size $I\times F$, $J\times F$, and $K\times F$, and $\mathbf{X}^{(I\times JK)}$ is the matrix of size $I\times JK$ obtained by unfolding the array $\mathbf{X}$ of size $I\times J\times K$ in the first mode. There exist several algorithms that fit the PARAFAC model. We focus on the most widely used among all: the ALS algorithm. ALS consists of estimating one of the three matrices at each step by minimizing in the least squares sense the error

$$\Upsilon = \| \mathbf{X}^{(I\times JK)} - \mathbf{A}(\mathbf{C}\odot\mathbf{B})^T \|_F^2,$$

where $\| \bullet \|_F$ denotes the Frobenius norm. With matrices $\mathbf{B}$ and $\mathbf{C}$ fixed to initial values, the estimate of $\mathbf{A}$ in the least square sense is given by

$$\widehat{\mathbf{A}} = \mathbf{X}^{(I\times JK)}(\mathbf{Z}_a^+)^T, \tag{2}$$

where $\mathbf{Z}_a = \mathbf{C}\odot\mathbf{B}$ and $(^+)$ is the Moore–Penrose pseudoinverse. We estimate matrices $\mathbf{B}$ and $\mathbf{C}$ in an equivalent way, with $\mathbf{Z}_b = \mathbf{A}\odot\mathbf{C}$ and $\mathbf{Z}_c = \mathbf{B}\odot\mathbf{A}$, and repeat the same steps until a convergence criterion is reached—typically when the error Υ exhibits, between two iterations, a change smaller than a predefined threshold, which varies depending on the data. For simple data it can be set to 10^{-6}, for example, but it should be smaller for difficult data, 10^{-10}, for example. Note that, in order to avoid a threshold that is scale dependent, a relative error can be used instead, or array $\mathbf{X}$ can be prenormalized by its Frobenius norm.

We summarize the steps of the ALS algorithm in Figure 1.

It sometimes happens that the convergence needs a very large number of iterations. Choosing good starting values will, in some cases, help to reach the global minimum very quickly. Sometimes, however, it is impossible to reach a global minimum quickly by ALS from any starting point because the solution is embedded in a deep swamp, or is in fact unreachable at the solution rank, and can only be approached through an infinite series of diverging better fitting sets of loadings, as described by Kruskal.

3. Line search. Bro (in [2, p. 95–96]) and Harshman (in [9, p. 32–33]) have pointed out the important fact that, when the convergence is slow, there exist cycles of convergence defined by a unique direction. Within a given cycle, the loading factors evolve in the same direction to the final solution of that cycle. The following cycles exhibit the same scenario. The convergence within the cycle can take several iterations. To limit the number of iterations of a given cycle, Harshman and Bro propose

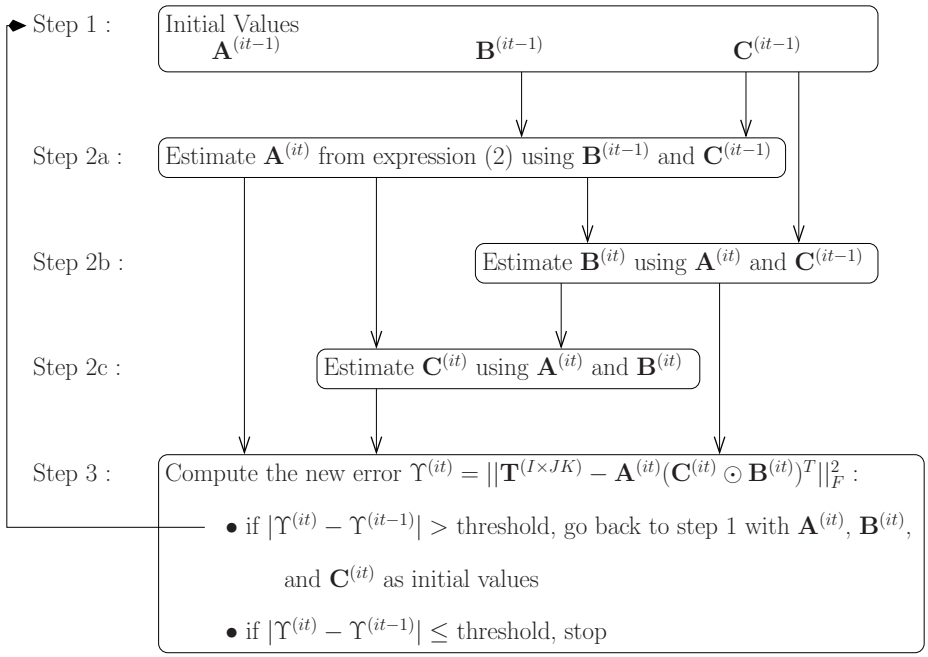


FIG. 1. *Steps of the ALS algorithm.*

to extrapolate, or more precisely, they propose to predict the value of the loading factors a certain number of iterations ahead by computing a sort of linear regression:

$$\mathbf{A}^{(new)} = \mathbf{A}^{(it-2)} + R_{LS}(\mathbf{A}^{(it-1)} - \mathbf{A}^{(it-2)}) \tag{3}$$

$\mathbf{A}^{(it-1)}$ is the estimate of matrix $\mathbf{A}$ obtained in the ALS iteration $(it-1)$, and $\mathbf{A}^{(new)}$ is the matrix that will be used in the itth iteration instead of $\mathbf{A}^{(it-1)}$. $(\mathbf{A}^{(it-1)} - \mathbf{A}^{(it-2)})$ defines the direction of the cycle. Matrices $\mathbf{B}^{(new)}$ and $\mathbf{C}^{(new)}$ are obtained in an equivalent way using the same relaxation factor R_{LS}. Of course, extrapolation should be very simple and does not make sense if it requires more time than the corresponding iterations. The simplest case is, of course, when R_{LS} is given a fixed value (between 1.2 and 1.3) [9], or is set to $it^{1/3}$ [2].

At every iteration it, the "new" loading factors are used to compute the error

$$\Upsilon^{(new)} = \| \mathbf{X}^{(I\times JK)} - \mathbf{A}^{(new)}(\mathbf{C}^{(new)} \odot \mathbf{B}^{(new)})^T \|_F^2 . \tag{4}$$

If $\Upsilon^{(new)} \geq \Upsilon^{(it-1)}$, then this means that we went too far in the extrapolation because R_{LS} is too large; R_{LS} is decreased, and we take the loading factors of iteration $(it-1)$ instead of the "new" ones. However, if $\Upsilon^{(new)} < \Upsilon^{(it-1)}$ then acceleration is accomplished and we gain some iterations.

The steps of the ALS algorithm with line search, as proposed by Andersson and Bro in [1], are summarized in Figure 2. The dashed area corresponds to the line search part.

Line search is executed after a few iterations of the ALS algorithm in order to wait for the system to stabilize. In [1] "few" is set to 6 but it could be higher depending on the data. The relaxation factor R_{LS} is defined for iteration (it) by : $R_{LS} = it^{1/n}$, with n fixed to 3 at the beginning of the simulation. When the acceleration fails several times (5 times in [1]), R_{LS} is decreased to $it^{1/(n+1)}$ and $\mathbf{A}^{(it-1)}$, $\mathbf{B}^{(it-1)}$, and $\mathbf{C}^{(it-1)}$

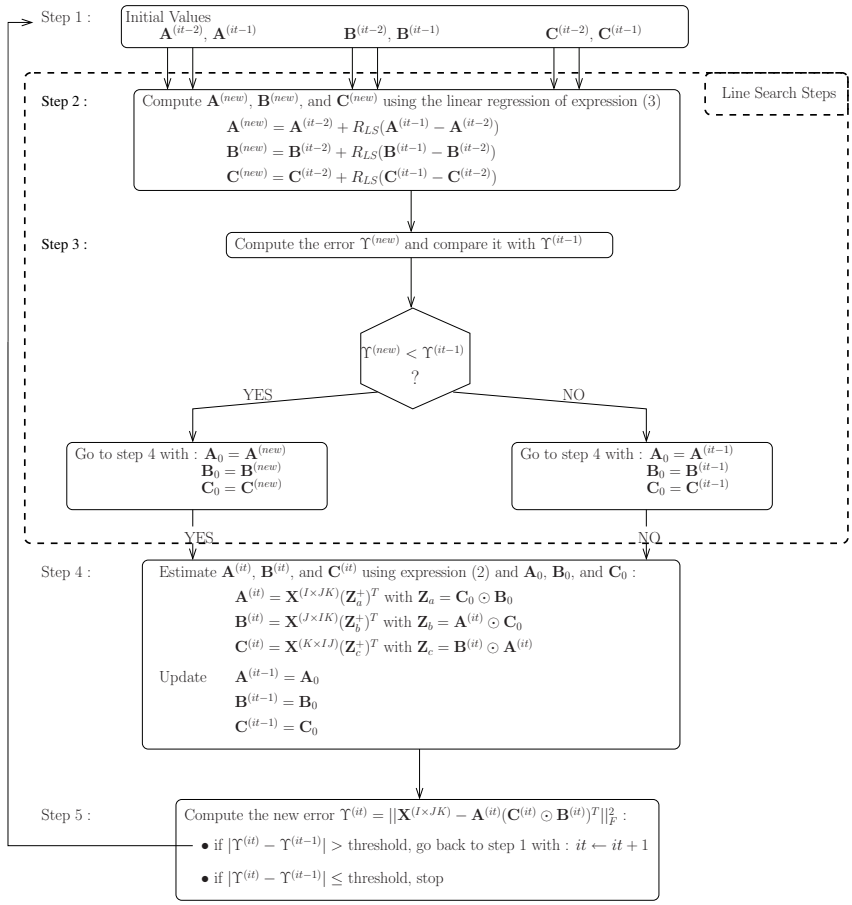


FIG. 2. *Steps of the ALS algorithm with LS.*

are used to update the loading factors of the current iteration (it) as described by the graph in Figure 2 at the end of the third step. However, when $\Upsilon^{(new)} < \Upsilon^{(it-1)}$, matrices $\mathbf{A}_0$, $\mathbf{B}_0$, and $\mathbf{C}_0$ are set to $\mathbf{A}^{(new)}$, $\mathbf{B}^{(new)}$, and $\mathbf{C}^{(new)}$, respectively. After estimating the loading factors at step 4, we update the loading matrices of iteration $(it-1)$ to $\mathbf{A}_0$, $\mathbf{B}_0$, and $\mathbf{C}_0$, and use them with those of iteration (it) for the next iteration (unless the algorithm has converged).

The fact that R_{LS} has a small value would suggest that the acceleration is not very efficient. This is not true since the effect of R_{LS} is compounded from one iteration to the next, leading eventually to a noticeable reduction of the number of iterations, as shown in Figure 10.

This linear extrapolation was applied to our synthetic data in section 5.6, as a basis for comparison with our ELS method. We will see that it can produce a clear improvement, for example reducing iterations in one example from about 10,000 to 5,000. However, since the number required is still high, it is still of interest to look for a novel method to reduce the number of iterations even further.

4. Enhanced line search (ELS). The idea of the enhanced line search (ELS) consists of seeking the optimal relaxation factor R_{LS} that leads to the final solution of a given cycle in only one step. For iteration (it), define $\mathbf{G}_a^{(it)} = \mathbf{A}^{(it-1)} - \mathbf{A}^{(it-2)}$ as the

direction of the cycle for loading matrix $\mathbf{A}$. $\mathbf{G}_b^{(it)}$ and $\mathbf{G}_c^{(it)}$ are defined equivalently. Instead of fixing a single value R_{LS} for the three modes as in expression (3), we may look for the optimal triplet (R_a, R_b, R_c) that minimizes

$$\begin{aligned}\Upsilon_{ELS} =& \| \mathbf{X}^{(I\times JK)} - (\mathbf{A}^{(it-2)} + R_a\mathbf{G}_a^{(it)}) \\ & \left((\mathbf{C}^{(it-2)} + R_c\mathbf{G}_c^{(it)}) \odot (\mathbf{B}^{(it-2)} + R_b\mathbf{G}_b^{(it)})\right)^T \|_F^2 \,. \end{aligned} \tag{5}$$

ELS is performed at the beginning of the ALS algorithm as shown in Figure 3, where step $1'$ corresponds to the ELS part. Relaxation factors applied to the loadings are no longer fixed as for the line search method in Figure 2, but they are computed at step $1'$ of Figure 3 as the optimal values that provide the smallest error Υ_{ELS}. At step 3, after estimating the loading matrices of iteration (it), we update those of iteration $(it-1)$ to $\mathbf{A}^{(new)}$, $\mathbf{B}^{(new)}$, and $\mathbf{C}^{(new)}$. Loading matrices of both iterations $(it-1)$ and (it) will then be used in the next iteration if the algorithm does not reach convergence.

The optimal solution is obtained when we jointly minimize Υ_{ELS} w.r.t. the three different factors R_a, R_b, and R_c. In this case the problem consists of solving a system of three polynomials in three unknowns, which leads to a high numerical complexity. Solutions with a smaller complexity are obtained by taking only two unknowns, or the same factor for all the modes $R = R_a = R_b = R_c$. Some of the possible optimizations are listed below:

- (R_a, R_b, R_c) which gives the optimal solution and involves a polynomial in three unknowns of degree 6.
- (R, R, R_c) where we use the same factor for $\mathbf{A}$ and $\mathbf{B}$ and we minimize Υ_{ELS} w.r.t. two variables R and R_c. This involves a polynomial in two unknowns of degree 6.
- (R, R, R) where we use the same factor for all matrices and involves a polynomial in a single unknown of degree 6.
- $R(R_b, R_c)$ where we use the relaxation factor of line search $R = it^{1/3}$ for matrix $\mathbf{A}$, and minimize (5) w.r.t. R_b and R_c. This involves a polynomial in two unknowns of degree 4.
- $R(R, R)$ which is the same as $R(R_b, R_c)$ with $R_b = R_c$, and involves a polynomial in a single unknown of degree 4.
- $R, R(R)$ where we optimize only w.r.t. to R_c.

In this article, the exploration of alternative ELS models is initiated by implementing (R, R, R), which is the simplest one that is "fully ELS." In this case, the error Υ_{ELS} is a polynomial of degree 6 in R (we omit the iteration index (it) to simplify the notation):

$$\begin{aligned}\Upsilon_{ELS}(R) &= \sum_{ijk}\left(X_{ijk} - \sum_{f=1}^{F}(A_{if} + RG_{a,if})(B_{jf} + RG_{b,jf})(C_{kf} + RG_{c,kf})\right)^2 \\ &= \sum_{d=0}^{6} p_d R^d, \end{aligned} \tag{6}$$

where p_d, $d = 0, \ldots, 6$ are functions of observed values stored array $\mathbf{X}$ and coefficients

of loading matrices of iterations $(it-1)$ and $(it-2)$; the expression of p_d are given in section A.2. To find the optimal R it suffices to determine the roots of polynomial $\Upsilon'_{ELS}(R)$, which provides five possible values of R. We feed those values into expression (6) and keep the one that gives the smallest error Υ_{ELS}.

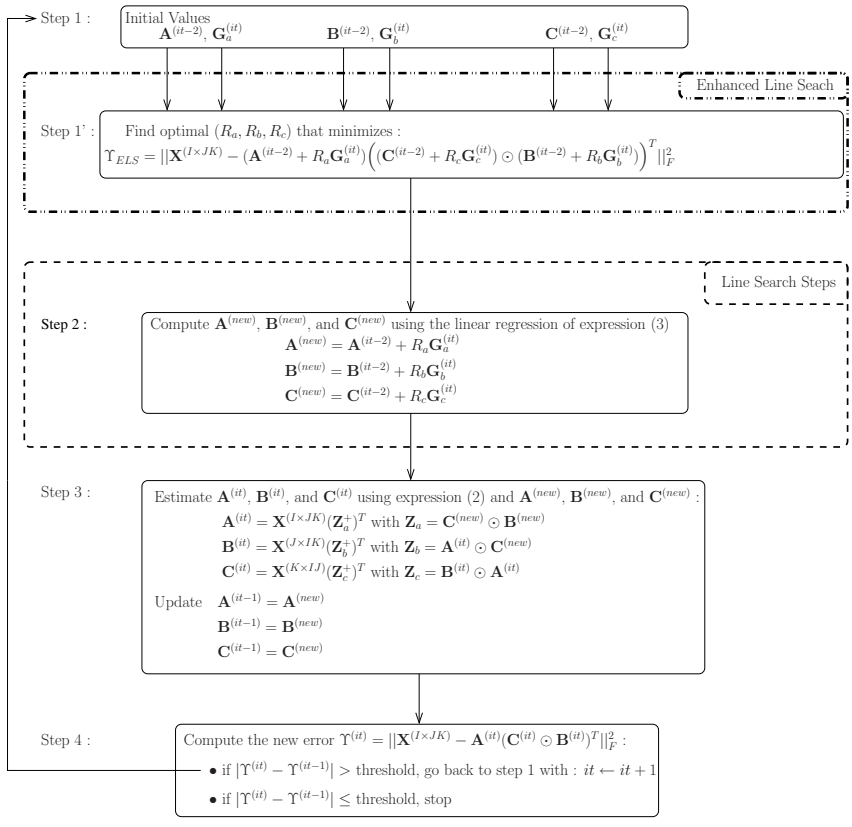


FIG. 3. *Steps of the ALS algorithm with ELS.*

To obtain some insight into whether the extrapolation is likely to be advantageous in the short-range sense (cf. question (i) posed at the end of this section), we can estimate the relative computation required by a single ELS iteration compared to LS. To do this, we compute the complexity of ALS and compare it with the complexity of optimization (R, R, R), for example. At each ALS iteration the following steps are performed:

1. Compute the optimal relaxation factor R by minimizing expression (5). To do so, take the derivative of (5) w.r.t. R, and root the obtained polynomial of degree 5 in one unknown.

2. Compute the new loading factors as in (3) and compute the corresponding error Υ_{new} given by expression (4).

3. Use $\mathbf{A}^{(new)}$, $\mathbf{B}^{(new)}$, and $\mathbf{C}^{(new)}$ as starting values for the PARAFAC iteration instead of $\mathbf{A}^{(it-1)}$, $\mathbf{B}^{(it-1)}$, and $\mathbf{C}^{(it-1)}$, and estimate the first loading factor $\widehat{\mathbf{A}}$ as shown in (2).

4. Perform step 3. To estimate each of the remaining loading factors $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{C}}$, by using matrices $\widehat{\mathbf{A}}$ and $\mathbf{C}^{(new)}$ to estimate $\widehat{\mathbf{B}}$, and matrices $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{B}}$ to estimate $\widehat{\mathbf{C}}$.

According to the details given in section A.1, one (ALS+ELS) iteration corresponds to about $(F + 7F^2)(IJ + JK + IK) + 3IJKF + 11F^3 + 2F^2(I + J + K) + (8F + 10)IJK$ multiplications. For instance, this equals 2601 multiplications, when $I = 2$, $J = 3$, $K = 3$, and $F = 3$. Without ELS, ALS requires $(F + 7F^2)(IJ + JK + IK) + 3IJKF + 11F^3 + 2F^2(I + J + K)$, which means 1989 multiplications in the same conditions. On the other hand, ELS makes the number of iterations decrease from 7100 to 250 iterations, hence allowing a decrease of the overall complexity from 14121900 to 650250 multiplications.

It is worth noting that $\Upsilon^{(new)}$ is always smaller than $\Upsilon^{(it-1)}$ when we use optimal values for R_a, R_b, and R_c as is the case for the first three optimizations. However, when we use a fixed relaxation factor as in LS, $\Upsilon^{(new)}$ can exceed $\Upsilon^{(it-1)}$, which means that the acceleration may fail.

This can explain the fact that, in theory, a single iteration of ELS should always improve fit as much as and almost always more than LS at any given point in the solution space. The questions then become: (i) Does the fit improvement turn out to be more beneficial than the cost of added computation is detrimental? (ii) Does the method find a significantly better path to the solution? Question (ii) is particularly important in cases where progress becomes very slow because of local characteristics of the hypersurface along which the fitting procedure is moving. Question (ii) is much subtler than (i). For example, one can easily imagine that a good long-range method might trade off some locally slower steps for a much better path a bit further along (a shortcut just over the horizon). This sort of question can only be answered well through application of a method to simulated and/or real data.

5. Computer results. To compare the performance of ELS to LS, a standard PARAFAC program was minimally modified to change the line search to ELS and to record time and fit information on each iteration. The test datasets were three-way and four-way synthetic data arrays, constructed according to the PARAFAC model to have specific kinds of factor structure and levels of random error. Then, in each test, the two algorithms were given identical problems; that is, they were given the same synthetic datasets, with the same analysis options, and started from the same random starting positions. This allowed us to compare the progress of the two methods step-by-step as they proceeded from a given starting point toward the best least-squares solution.

To obtain a general picture of the relative performance of ELS and LS, we considered a wide range of datasets[1]. A fully systematic exploration has not yet been completed, and even the partial results obtained so far require much more space than is available for this article, so we present here summary conclusions for each of our main test conditions, and give a few illustrative examples.

In most experimental conditions (i.e., most of the data structure types tested), the iteration count for ELS was substantially lower than that for LS, and in no condition was it (reliably) higher. This is consistent with the theoretical expectations described earlier. On the other hand, ELS *execution times* were longer than LS by

[1]We took as our "standard" the PARAFAC function contained in Andersson and Bro's [1] N-way Toolbox for MATLAB (which may be found at http://www.models.kvl.dk/source/nwaytoolbox/index.asp). The same MATLAB code was used except that the ELS extrapolation code replaced the LS extrapolation code in the function. In both versions of the PARAFAC function, a few lines were inserted that obtained time information on each iteration and saved it along with the current value of fit computed by the program at that iteration. Release 7 of MATLAB was used in all experiments.

more than we expected—generally the ratio of ELS to LS execution time per iteration was always the same for any given problem, but was larger than we would predict from our approximate estimates of complexity computation; despite streamlining and vectorization of the code, the ELS to LS ratio was still somewhat larger than our complexity estimates by roughly a factor of three. This might be due to inefficiencies in our ELS algorithm, or perhaps some particularly efficient features in the LS code, or as yet unanticipated considerations. Since reduction in the execution time is the goal of the proposed method, we feel that it is important to communicate both the time and iteration reductions we have observed, even though we are not yet sure of how to interpret the time information (or even whether or not it is somehow artifactual). We will describe a theoretical adjustment that brings times in line with computational complexity, and this provides one way of dealing with the current uncertainty in our timing results.

Complexity differences. In applications where the rank F is small in the sense of inequality (7) given in the appendix, the (R, R, R) version of ELS will significantly increase the complexity per iteration, and hence the CPU time required per iteration over that of LS. Whether or not ELS is attractive thus depends in part on the problem size and dimensionality.

Datasets sizes and numbers of factors to be extracted from them vary from one discipline to the next. In chemometrics and signal processing, typical problems might involve data arrays of the order of $60 \times 60 \times 20$ and perhaps three or four factors to be extracted. In such cases, the computational complexity of ELS is approximately three times that of LS, at least as estimated by the formulae in the appendix. For these problems, use of ELS would be attractive only for classes of problems in which it is clearly superior to LS in ability to traverse the curvature of the solution space. To provide a benefit beyond the use of LS, the ELS method needs to reduce the number of iterations required by LS.

It turns out that a single "bottleneck"—one of the most common and simple kinds of convergence slowdown—appears to have the required properties. We have also found other classes, such as triple bottleneck, where ELS will actually increase substantially the time needed to find the solution.

The data variation that turned out to have the most important impact on the relative performance of the two methods was the factor correlation structure both within and across modes. When no modes had collinear factors and all factors were of roughly equal size, there was no convergence difficulty for either method. For example, midsized datasets (e.g., $45 \times 40 \times 35$) often satisfied the convergence criterion—usually a change in root mean square error (RMS) of 10^{-8} between successive iterations—in 15–75 iterations. ELS usually took fewer iterations to converge, but the iterations were slower. In other words, with our MATLAB implementation, ELS often took on the order of 15%-25% *more time* than LS to reach the convergence criterion, even if figures given by computational complexity calculations are more optimistic.

5.1. Dealing with bottlenecks. We have considered several different situations where convergence of ALS PARAFAC algorithms become slow, but have focused mainly on the one that is the simplest and (outside of the social sciences) the most common: simple factor collinearity. This has several different versions; some or all factors can be collinear in one or several factor loading matrices that define the variation structure of an array. We only briefly look at the other important case—the more complicated kind of convergence difficulties caused by "degenerate PARAFAC solutions."

When one of the factor matrices in the optimal solution has two or more collinear columns, resolving them can seriously slow down the overall progress of ALS estimation of the factors, even though the solution may eventually be well defined. Harshman terms this situation a "bottleneck" [12]. When such collinearity is present in two or

three modes of the array (i.e., two or three of the "latent" factor loading matrices), then one has a structure with a "double" or "triple" bottleneck. We created synthetic data involving single, double, and triple bottleneck structure to test the performance of ELS vs. LS in these conditions. The (R, R, R) version of ELS that we used for these tests behaved quite differently in single vs. multiple bottleneck situations—at least for three-way arrays.

5.2. Single bottleneck situations: When factors in one mode are collinear. In our tests, ELS always outperformed LS when only one mode had factor loading vectors that were almost collinear, providing the analysis reached a global optimum in which all factors were approximately recovered. The time and iteration values observed in one such run are shown in Figure 4.

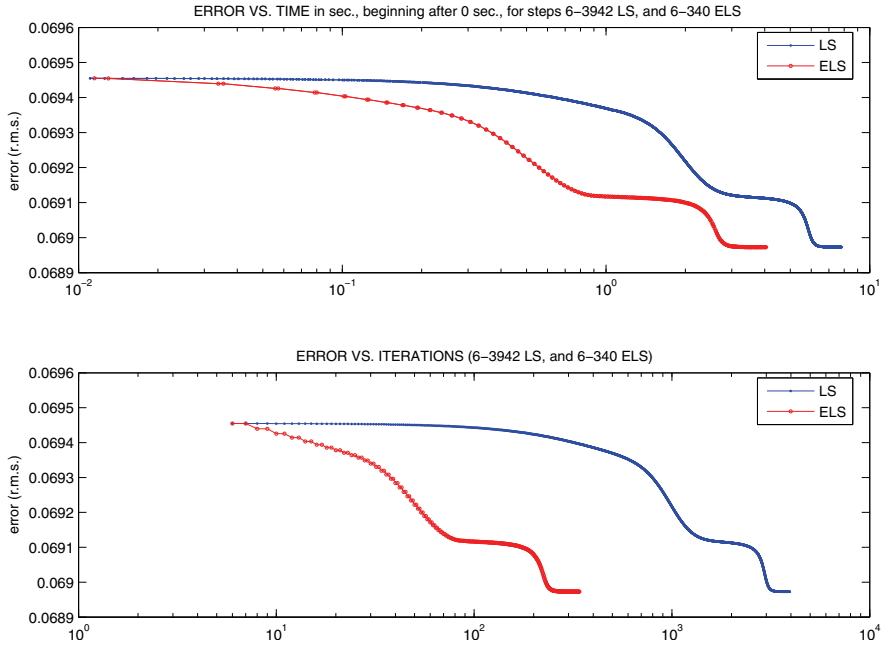


FIG. 4. *Performance of ALS with both LS or* (R, R, R)*-ELS accelerations, on a* $12 \times 11 \times 10$ *array of rank* 5*, in the presence of a single bottleneck (*3 *factors out of* 5 *are almost collinear in one of the* 3 *modes).*

The example shown is for an array with 5 factors and the lowest level of random noise used in the tests (0.1%). Three of the factors were almost collinear (separated by 10 degrees in the Mode C factor space) while the other two were roughly orthogonal to the other factors. The shape of the curves is quite similar. This suggests that the two algorithms are following "similar" or somewhat parallel paths through the solution space, and are encountering a similar sequence of more and less difficult regions in the solution space, but they are progressing at different rates because ELS tends to make bigger improvements in fit. As shown here, ELS often reduced the total number of iterations by approximately an order of magnitude, but because its iterations took somewhat longer (at least with our MATLAB implementation), the time reduction was between half and two-thirds of the size of the reduction in iterations. The fit and time curves in Figure 4 are based on a relatively small dataset ($12 \times 11 \times 10$ with five factors). In that case, index (7) is 1.37, which shows that an ELS iteration is more complex than LS.

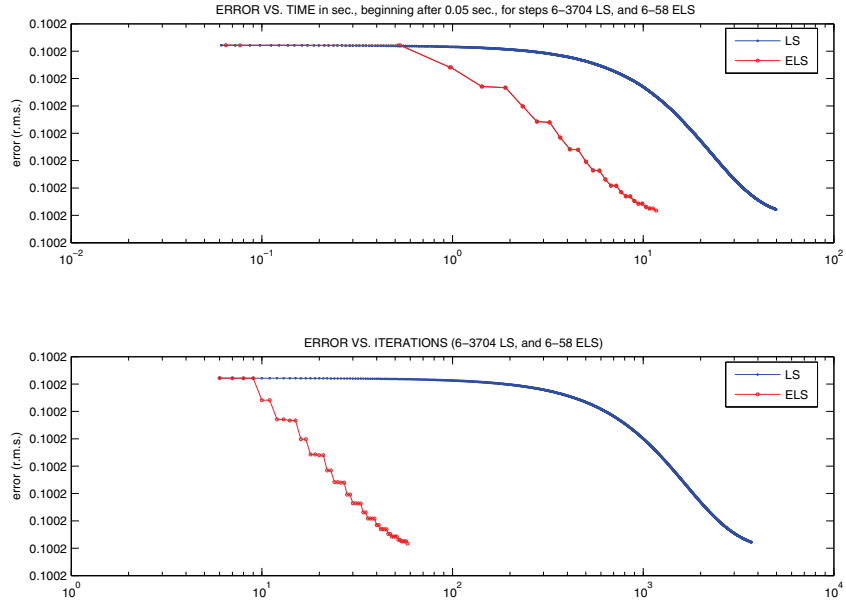


FIG. 5. *Performance of ALS with both LS or* (R, R, R)*-ELS accelerations, on a* $80 \times 60 \times 20$ *array of rank* 5*, in the presence of a single bottleneck.*

With the larger dataset considered next, index (7) is even smaller, and the relative computational complexity of ELS is at least three times that of LS per iteration. Our MATLAB implementation takes closer to 9 times as much time, for reasons that we are not yet able to fully explain. ELS is, however, still attractive in this low-rank large-dimension case, despite the relatively high complexity per iteration.

Note the clear two-step pattern in Figure 5, both in the fit drop and in the associated time needed at each successive iteration (i.e., two-iteration values are close to one another, and then there is a larger interval, and two are close to one another again). The pattern is present in both curves, but is much more obvious in the ELS curve because of the rapid drop in fit at every other step. This "paired-step" pattern is due to the way LS is currently implemented in the PARAFAC function for the N-way toolbox. The program collects two sets of loadings, from two successive iterations, and then extrapolates based on those two, and collects two more, etc. Thus the extrapolation occurs on every other iteration. Because ELS was incorporated into the exact same loops that governed LS, it too is applied on every other iteration.

Our tests indicate that LS is considerably more effective when the extrapolation is performed on every iteration, as in the original extrapolation used in Harshman 1970 [9]. However, we concentrate in this article on a direct comparison of paired-step LS with paired-step ELS (we have also begun some comparisons between the two methods when both are performed at every iteration, and our preliminary results suggest that in this case the performance difference between them is smaller).

For this dataset, as with all other "single bottleneck" cases we have tested, paired-step ELS clearly outperforms paired-step LS—so long as they find the global optimum. However, when the path taken by an analysis traps it in a local optimum, or when some of the highly collinear factors are too poorly resolved due to error in the data, the behavior of LS and/or ELS changes and the time advantage offered by ELS is reduced or eliminated. ELS continues to require fewer iterations, but the difference between the counts becomes small enough that ELS does not reduce the overall time

(at least with our MATLAB implementation).

5.3. Multiple bottlenecks and degenerate solutions. Multiple bottlenecks and degenerate solutions appear when experimental requirements or practical limitations in data collection make collinearity of some factors unavoidable; this most commonly applies to only one mode of the data array. The previous results are for single-mode bottlenecks which leads us to the tentative conclusion that for such cases ELS would seem an attractive estimation method. There are situations, though, in which some subset of the factors will be collinear in two modes, or even in all three modes of a three-way array. Our experiments therefore simulated these (less common) kinds of data as well. We found that the advantage of ELS does not generally extend to these situations. The curvature of the path to the optimum gets more complicated and apparently makes ELS "shortcut" methods less successful. This is another demonstration of the subtlety of the considerations involved in nonlinear extrapolation of PARAFAC solutions.

Neither double- nor triple-bottleneck situations benefited much from the simplest (R, R, R) version ELS. In general, the *time* required by our MATLAB implementation to reach the converged solution was increased. However, our test cases often were hard to fit, so we had to take care to distinguish global optimum cases from local optimum ones. To obtain global optima with adequate frequency in the double-bottleneck case, the angle between factors had to be increased to moderate values (25 degrees in the case of Figure 6), making the collinearity in individual factor spaces less extreme but the combined effects of the collinearity in the two or three modes was still fairly severe. Unfortunately, even in clear cases of having reached the global optimum, where recovery of all factors was close to perfect (when the noise level was set at 0.001), the ELS method usually took longer (in CPU time) to converge than simple LS. Figure 6 shows the results of one such triple-bottleneck case.

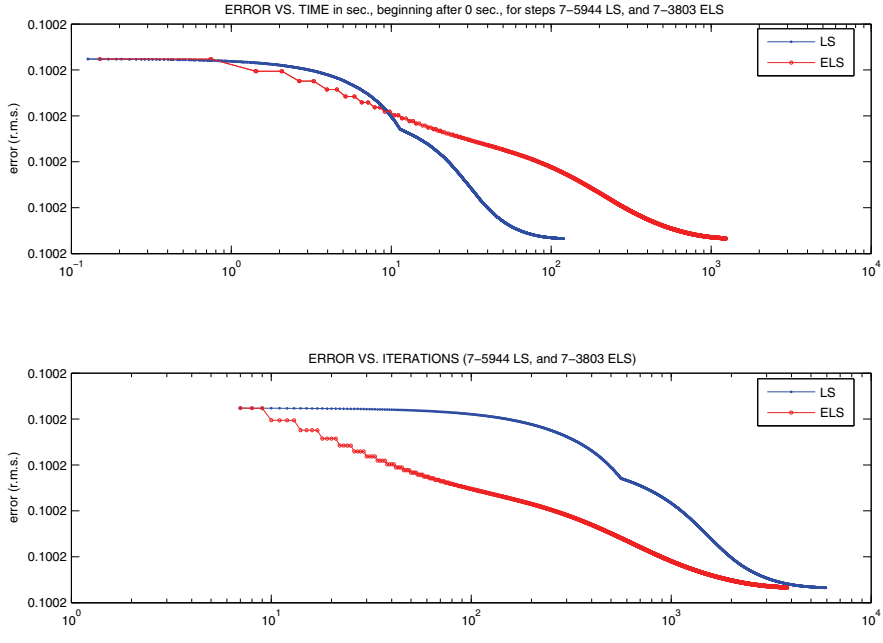


FIG. 6. *Performance of ALS with both LS or* (R, R, R)*-ELS accelerations, on a* $80 \times 60 \times 20$ *array of rank* 4*, in the presence of a triple bottleneck.*

5.4. Degenerate and quasi-degenerate solutions. There is another important situation in which serious convergence difficulties arise. This happens when the factors are not particularly collinear on average, but the angle between them varies greatly across levels of the array (i.e., differs across values of the third array index). This kind of variation in factor structure is not consistent with the PARAFAC model. However, PARAFAC can fit part of the "axis wobble" or "Tucker Variation" [14] by reweighting axes after the space has been sheared [11]. Thus, when too much axis "wobble" is present, the **A**, **B**, and **C** factor spaces become inversely sheared to better fit it, creating a "degenerate solution," which involves strong collinearities and seriously impedes convergence. Degenerate solutions are usually dealt with by imposing constraints. However, speedup methods like ELS could be useful if they accelerated progress through "swamps" or deep into a swamp to find a final solution. In our initial tests of the (R, R, R) method, ELS has not been helpful in dealing with swamps, but it is not unreasonable to conjecture that versions more sophisticated than (R, R, R), as suggested in section 4, might do better in these situations.

5.5. A four-way example with two collinear and two noncollinear modes. From the experiments reported so far, it is unclear whether the relative lack of ELS success when applied to factor structures with double and triple bottlenecks is because of too many bottlenecks or too little "wiggle room." That is, we cannot distinguish the cases that have multiple modes with bottlenecks from those that do not have multiple modes without bottlenecks. In the present section, we report some earlier studies on the impact of ELS on ALS in four-way arrays. In these arrays, two modes have almost collinear factors and two do not. If multiple bottlenecks is what creates the convergence problem, then ELS should also encounter difficulties in these datasets. If lack of "wiggle room" is what creates the convergence problem, then ELS should be better at dealing with double bottlenecks.

The experiments we are about to describe are simpler in two important ways: (a) the tests did not measure or record *execution time* information; (b) the datasets were constructed as error-free arrays, that is, without adding any random noise. The first limitation makes the interpretation difficult, but the dramatic reduction in iteration counts does appear impressive when compared to the relatively modest differences in iterations in, for example, Figure 6. The second limitation can be minimized by qualifying our interpretation. The difference is nontrivial because when collinearities are combined with error, it can complicate the algorithm's task of resolving the highly similar factor profiles. However, the results are still informative if considered as demonstrating certain theoretical/mathematical properties of four-way (R, R, R)-ELS. They might also be interpreted as simulations of real world cases where the error is sufficiently small to make the behavior of the algorithms roughly equivalent to those found in these error-free cases.

We consider the four-way PARAFAC model:

$$X_{ijkl} = \sum_{f=1}^{F} A_{if} B_{jf} C_{kf} D_{lf},$$

where

$$\mathbf{A} = \begin{pmatrix} 1 & \cos(\theta) & 0 & \sin(\theta) \\ 0 & \sin(\theta) & 1 & \cos(\theta) \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 3 & \cos(\theta) & 0 & \sin(\theta) \\ 0 & \sin(\theta) & 1 & \cos(\theta) \\ 0 & \sin(\theta) & 0 & \sin(\theta) \end{pmatrix},$$

and $\mathbf{C}$ and $\mathbf{D}$ are randomly generated matrices of size 3×4. The collinearity is controlled through variable θ. We take $\theta = \pi/60$ in Figures 7 and 8. The first and second columns of each of the matrices $\mathbf{A}$ and $\mathbf{B}$ are almost collinear as θ is very close to zero ($\theta \simeq 0.052$). The same thing holds for the third and fourth columns of $\mathbf{A}$ and $\mathbf{B}$.

This example demonstrates one of the cases where results of [25] and [21] cannot be applied, since there are more columns than rows in the loading matrices and so they do not have full column rank.

We notice from Figure 7 that ELS reduces the number of iterations needed to meet the criterion for approximate convergence from more than 10000 to about 2000! We report in Figure 8 the median of the loss function for one dataset, over 100 independent trials (with 100 different random initial values). We notice that even though ALS + LS reaches the error 10^{-4} very quickly, it is then trapped for many iterations ("trapped in the bottleneck"). In contrast, ALS + ELS escapes comparatively quickly from the bottleneck (after 1000 iterations) and converges to smaller values of the error 10^{-12}, while ALS and ALS+LS remain in the plateaux 10^{-4} and 10^{-5}, respectively.

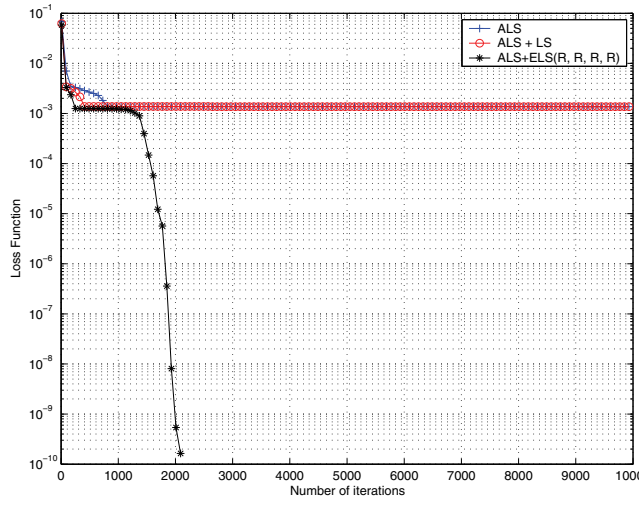


FIG. 7. *Loss function* Υ *as a function of the number of iterations for ALS with LS, and ELS with optimization* (R, R, R, R) *for* $\theta = \pi/60$.

Figure 9 appears to confirm this hypothesis by showing more frequent changes in parameter sets. It gives the variation of the coefficients of matrix $\hat{\mathbf{A}}$ as a function of the number of iterations. During progress through a bottleneck, the variation of $\hat{\mathbf{A}}$ coefficients is very small and this increases when we get out of the bottleneck. The same thing is evident in matrices $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$.

5.6. ELS applied to blind channel identification of an UDM. This second four-way example demonstrates an application of ELS to blind identification of an under-determined mixture (UDM). Specifically, we use ELS to accelerate ALESCAF, the algorithm proposed in [7] for blind channel identification based on the characteristic function in an UDM.

Using the notation defined in [7], ALESCAF leads to a four-way PARAFAC model:

$$\mathbf{T}^{(P \times KP^2)} = \mathbf{A}(\mathbf{D} \odot \mathbf{A} \odot \mathbf{A})^T.$$

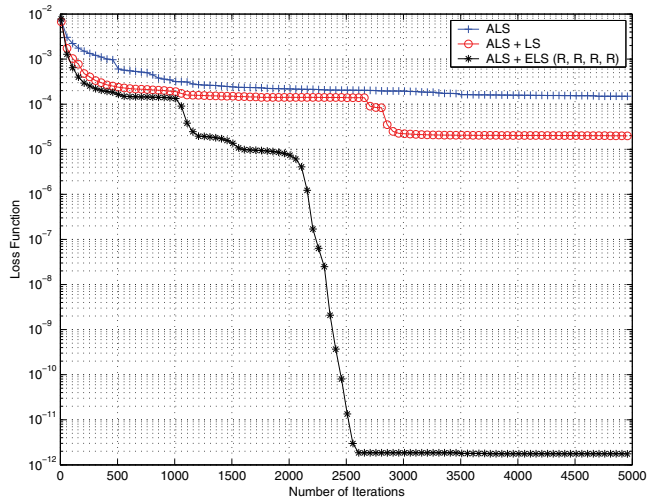


FIG. 8. *Loss function* Υ *as a function of the number of iterations for ALS with LS, and ELS with optimization* (R, R, R, R) *for* $\theta = \pi/60$*, median value over* 100 *independent trials.*

The array $\mathbf{T}$ contains the third derivatives of the joint characteristic function of the observations computed at K points of the grid Ω. Matrix $\mathbf{D}$ is obtained from the independence property of the sources and its entries are defined as

$$D_{kn} = \psi_n^{(3)} \left(\sum_q A_{qn} u_q[k] \right),$$

where $1 \leq k \leq K$ and $1 \leq n \leq N$. $\mathbf{A}$ is the channel matrix of size 2×3 to be identified.

As in our earlier tests, we use the MATLAB ALS implementation of PARAFAC made available by Andersson and Bro (http://www.models.kvl.dk) and described in [1]. Also as before, we create our ELS version by replacing their LS procedure by our ELS procedure, this time the (R, R, R, R) version. The three sources are BPSK, and we generate an "infinite block" of data by taking all of the 2^3 possible combinations of $\{-1, 1\}$, and we take 10000 as the maximum number of iterations. As in the previous four-way example, noise is not taken into account.

In Figure 10 we report the gap between estimated and actual mixing matrix using ELS and compare it with ALS with LS and nonaccelerated ALS. Figure 11 gives the error as a function of the number of iterations. The figure shows that ELS is very useful for reducing the number of iterations needed in three-bottleneck versions of four-way arrays. The number of iterations decreases from 5000 when using ALS with LS, to 500 when using optimization (R, R, R, R) of ELS. On the one hand, these results seem to be too dramatic to be "canceled out" by increases in iteration time, but on the other hand, the LS and ELS parameter changes (Figure 11) make us somewhat more cautious, since these seem more similar.

Overall, these four-way results encourage us to hope that when there is at least one mode that is free of collinearities, this type of ELS might be generally helpful.

A few cautionary points should be noted: When making comparisons of the methods, it might be wise to underemphasize the dramatic ELS drops of the error (or objective function) when the value falls below something like 10^{-3} or 10^{-4}, since these would be impossible with most real data containing measurement error or other disturbances of the data values. It is also not known how the behavior in the graphs

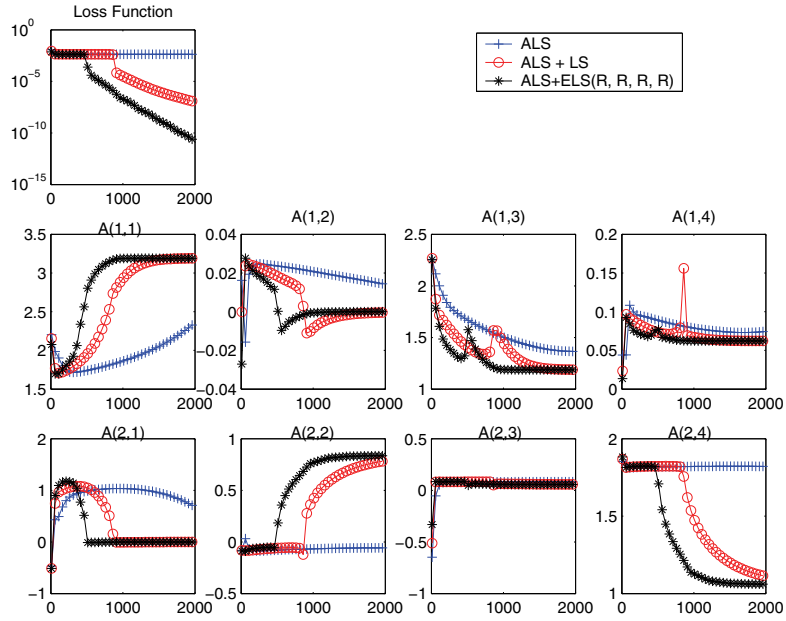


FIG. 9. $\hat{\mathbf{A}}$ *coefficients as a function of the number of iterations.*

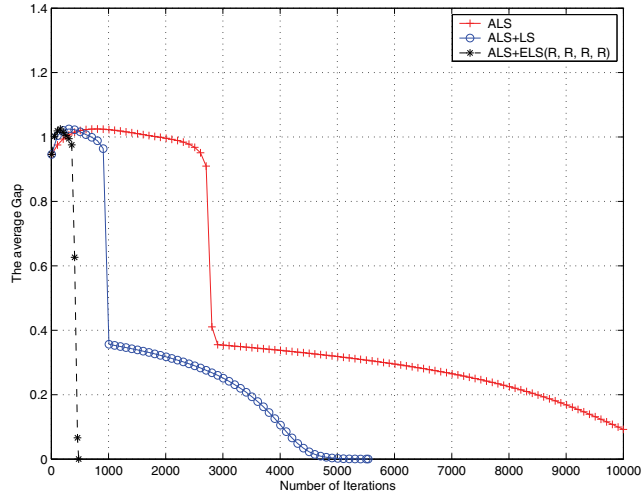


FIG. 10. *Gap between estimated and original channel matrix as a function of the number of iterations using ALS, ALS with LS, and ALS with ELS.*

associated with our four-way examples might be modified even at higher levels by the presence of random error; the three-way experiments suggest that some differences can be expected.

6. Concluding remarks. ELS is a novel technique aiming at accelerating convergence of the ALS algorithm when used to fit the PARAFAC model. Our simulations indicated that ELS could be a very attractive way to deal with "single bottleneck" situations—three-way arrays that have factor collinearities in one of the modes. As

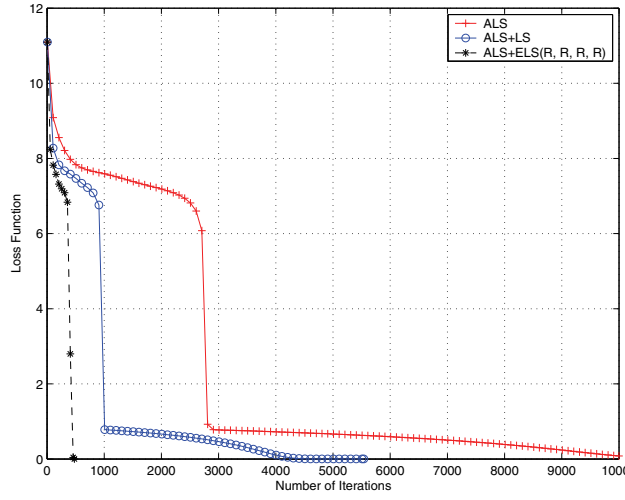


FIG. 11. *Loss function* Υ *as a function of the number of iterations using ALS, ALS with LS, and ALS with ELS.*

shown in Figures 4 and 5, ELS often decreased the number of iterations by an order of magnitude and the time to convergence by at least half to two-thirds of an order of magnitude (except when trapped in a local optimum). This was more than enough to counterbalance the longer iteration times due to the higher computational complexity.

On the other hand, in double- and triple-bottleneck three-way factor structures (i.e., where at most only one mode is free of collinearities), the (R, R, R) version of ELS that was tested here did not offer an advantage, but instead appeared to lengthen the overall time to convergence. However, we remain open to the possibility that our MATLAB time information is somehow unrepresentative or at least does not reflect what might be possible.

Two applications involving four-way arrays (with high rank) lacked information on execution time but some comparison based on computational complexity ratios could be a basis for tentative predictions. The encouraging reductions in iterations seen here open up the possibility that even when an array has three modes with collinearities, this is not necessarily a problem for ELS—if there is also one mode without factor collinearities. This possibility should be explored further.

This article presents some initial exploration of a nonlinear approach to ALS extrapolation, but the investigation is obviously a work in progress. Our theoretical understanding of multistep properties of ELS is still quite incomplete. We have identified some classes of problems where it works better than LS and others in which it may not. The dramatic improvements that can be obtained by ELS in "single bottleneck" situations is well demonstrated. And even if the (R, R, R) implementation of ELS reaches its limits and does not perform significantly better than LS in the absence of two noncollinear modes, or in the presence of convergence swamps arising from "degenerate PARAFAC solutions," there may be other implementations of ELS, such as that called $R(R, R)$ in section 4, which could be more successful in these circumstances. These issues remain to be studied.

Appendix A.

A.1. ELS steps and complexity. During one iteration of the ALS algorithm, the following operations are performed (we list the operations for the estimation of

$\widehat{\mathbf{A}}$, but similar operations are required for the two other loading factors):

1. Compute the Khatri–Rao product to obtain matrix $\mathbf{Z}^a$. This costs FJK multiplications.

2. Compute $\mathbf{Z}_a^+$ by reduced SVD of $\mathbf{Z}_a$, which requires $7JKF^2 + \frac{11}{3}F^3$ multiplications.

3. Estimate the factor loading $\widehat{\mathbf{A}}$ as shown in expression (2), which requires $IJKF + IF^2 + IF$ multiplications (if we assume $F \leq JK$).

As a consequence, the whole ALS iteration for the 3 modes requires an order of $(F + 7F^2)(IJ + JK + IK) + 3IJKF + 11F^3 + 2F^2(I + J + K)$multiplications.

Now let us evaluate the additional computational complexity involved by ELS. In order to do this, note that the ELS criterion can be rewritten as

$$\Upsilon_{ELS} = \sum_{ijk} \left[Y_{ijk} + R\, D_{ijk} - R^2\, E_{ijk} + R^3\, F_{ijk} \right]^2 .$$

The explicit calculation of arrays $\mathbf{Y}$, $\mathbf{D}$, $\mathbf{E}$, and $\mathbf{F}$ requires an order of $8IJKF$ multiplications. Next, the calculation of the coefficients of the degree-6 polynomial in R requires $10IJK$ multiplications. The computation of stationary points and the selection of the absolute minimum yield a negligible complexity since of order $O(5^3)$. As a conclusion, the additional complexity generated by ELS is thus of order $(8F+10)IJK$ multiplications when we choose the optimization with respect to a single factor R, that is, (R, R, R) for a three-way PARAFAC model. This is not negligible and can be considered to be small only for large enough F and small enough dimensions. More precisely, if

$$F\left(\frac{1}{I} + \frac{1}{J} + \frac{1}{K}\right) \gg 1, \tag{7}$$

then the additional complexity required by ELS may be considered to be negligible over that of LS. For instance, this is the case of generic arrays of size $(I, J, K) = (10, 10, 10)$, which have a rank $F = 36$.

More generally, it is interesting to evaluate the computational complexity for N-way arrays of size $I_1 \times I_2 \times \ldots \times I_N$, when all of the dimensions are of the same order $O(I)$. For ALS we get $\frac{11}{3}NF^3 + 2F^2NI + NI^NF + 7F^2NI^{N-1} + FNI^{N-1}$. On the other hand, it can be shown that ELS requires $[2^N F + O(N^2)]I^N + O((2N-1)^3)$ additional multiplications.

A.2. Expression of p_d. We define $q_{f,d}$ for $d = 0, \ldots, 6$ as

$$\begin{aligned}
q_{f,0} &= A_{if}B_{jf}C_{kf}, \\
q_{f,1} &= A_{if}B_{jf}G_{c,kf} + A_{if}G_{b,jf}C_{kf} + G_{a,if}B_{jf}C_{kf}, \\
q_{f,2} &= A_{if}G_{b,jf}G_{c,kf} + G_{a,if}B_{jf}G_{c,kf} + G_{a,if}G_{b,jf}C_{kf}, \\
q_{f,3} &= G_{a,if}G_{b,jf}G_{c,kf}.
\end{aligned}$$

$q_{f,d}$ depends on i, j, k but we omit the indices for simplicity. Then polynomial coefficients p_d are given by

$$\begin{aligned}
p_0 &= \textstyle\sum_{ijk} \big(X_{ijk} - \sum_f q_{f,0}\big)^2,\\
p_1 &= -2\textstyle\sum_{ijk} \big(X_{ijk} - \sum_f q_{f,0}\big)\big(\sum_f q_{f,1}\big),\\
p_2 &= \textstyle\sum_{ijk} \big(\sum_f q_{f,1}\big)^2 - 2\big(X_{ijk} - \sum_f q_{f,0}\big)\big(\sum_f q_{f,2}\big),\\
p_3 &= 2\textstyle\sum_{ijk} \big(\sum_f q_{f,1}\big)\big(\sum_f q_{f,2}\big) - \big(X_{ijk} - \sum_f q_{f,0}\big)\big(\sum_f q_{f,3}\big),\\
p_4 &= \textstyle\sum_{ijk} \big(\sum_f q_{f,2}\big)^2 + 2\big(\sum_f q_{f,1}\big)\big(\sum_f q_{f,3}\big),\\
p_5 &= 2\textstyle\sum_{ijk} \big(\sum_f q_{f,2}\big)\big(\sum_f q_{f,3}\big),\\
p_6 &= \textstyle\sum_{ijk} \big(\sum_f q_{f,3}\big)^2.
\end{aligned}$$

REFERENCES

[1] C. A. Andersson and R. Bro, *The n-way toolbox for MATLAB*, Chemom. Intell. Lab. Syst., 52 (2000), pp. 1–4.

[2] R. Bro, *Multi-way Analysis in the Food Industry: Models, Algorithms, and Applications*, Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands, 1998.

[3] J. D. Carroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart–Young decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[4] R. B. Cattell, *Parallel proportional profiles and other principles for determining the choice of factors by rotation*, Psychometrika, 9 (1944), pp. 267–283.

[5] R. B. Cattell and A. K. S. Cattell, *Factor rotation for proportional profiles: Analytical solution and an example*, British Journal of Statistical Psychology, 8 (1955), pp. 83–92.

[6] P. Comon, *Blind channel identification and extraction of more sources than sensors*, in Proceedings of the SPIE Conference, San Diego, 1998, pp. 2–13. Republished in IEEE Trans. Signal Process, 52 (2004), pp. 11–22.

[7] P. Comon and M. Rajih, *Blind identification of under-determined mixtures based on the characteristic function*, in ICASSP'05, vol. IV, Philadelphia, 2005, pp. 1005–1008.

[8] A. Franc, *Etude Algébrique des Multitableaux: Apports de l'algèbre Tensorielle*, Ph.D. thesis, University of Montpellier II, Montpellier, France, 1992.

[9] R. A. Harshman, *Foundations of the Parafac procedure: Models and conditions for an explanatory multimodal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[10] R. A. Harshman, *Determination and proof of minimum uniqueness conditions for PARAFAC*1, UCLA Working Papers in Phonetics, 22 (1972), pp. 111–117.

[11] R. A. Harshman, *The Problem and Nature of Degenerate Solutions or Decomposition of* 3*-way Arrays*, 2004. Discussion presented at the American Institute of Mathematics Tensor Decompsition Workshop, Palo Alto, CA. Slides available online at http://publish.uwo.ca/~harshman.

[12] R. A. Harshman, *A Note on Several Different Kinds of PARAFAC Degeneracy and Other Ill-Conditioning*, 2007, unpublished.

[13] R. A. Harshman and S. Hong, *"Stretch" versus "slice" methods for representing three-way structure via matrix notation*, J. Chemom., 16 (2002), pp. 198–205.

[14] R. A. Harshman and M. E. Lundy, *The PARAFAC model for three-way factor analysis and multidimensional scaling*, in Research Methods for Multimode Data Analysis, H. G. Law, C. W. Snyder, J. Hattie, and R. P. McDonald, eds., Praeger, New York, 1984, pp. 122–215, also available online at http://publish.uwo.ca/~harshman/lawch5.pdf.

[15] R. A. Harshman and M. E. Lundy, *Conditions governing full and partial Parafac uniqueness when two loading matrices have full column rank*, submitted, 2007.

[16] J. B. Kruskal, *Three-way arrays: Rank and uniqueness of trilinear decompositions with applications to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.

[17] J. B. Kruskal, *Rank, decomposition, and uniqueness for* 3*-way and n-way arrays*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., Elsevier Science, North-Holland, 1989, pp. 7–18.

[18] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*, Ph.D. thesis, K. U. Leuven, E. E. Department -ESAT, Belgium, 1997.

[19] L. De Lathauwer, *A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666.

[20] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Computation of the canonical decomposition by means of simultaneous generalized Schur decomposition*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 295–327.
[21] S. E. Leurgans, R. T. Ross, and R. B. Abel, *A decomposition for three-way arrays*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1064–1083.
[22] P. Paatero, *The multilinear engine—A table-driven, least squares program for solving multilinear problems, including the n-way parallel factor analysis model*, J. Comput. Graph. Statist., 8 (1999), pp. 854–888.
[23] W. S. Rayens and B. C. Mitchell, *Two-factor degeneracies and a stabilization of PARAFAC*, Chemom. Intell. Lab. Syst., 38 (1997), pp. 173–181.
[24] R. T. Ross and S. Leurgans, *Component resolution using multilinear models*, Methods Enzymol., 246 (1995), pp. 679–700.
[25] E. Sanchez and B. R. Kowalski, *Tensorial resolution: A direct trilinear decomposition*, J. Chemom., 4 (1990), pp. 29–45.
[26] N. D. Sidiropoulos and R. Bro, *On the uniqueness of multilinear decomposition of n-way arrays*, J. Chemom., 14 (2000), pp. 229–239.
[27] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis with Applications in the Chemical Sciences*, Wiley, Chichester, UK, 2004.
[28] G. Tomasi, *Practical and Computational Aspects in Chemometric Data Analysis*, Ph.D. thesis, Royal Veterinary and Agricultural University, Frederiksberg, Denmark, 2006.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1148–1171

SENSITIVITY ANALYSIS FOR THE PROBLEM OF MATRIX JOINT DIAGONALIZATION*

BIJAN AFSARI†

Abstract. We investigate the sensitivity of the problem of nonorthogonal (matrix) joint diagonalization (NOJD). First, we consider the uniqueness conditions for the problem of exact joint diagonalization (EJD), which is closely related to the issue of uniqueness in tensor decompositions. As a byproduct, we derive the well-known identifiability conditions for independent component analysis (ICA) based on an EJD formulation of ICA. We next introduce some known cost functions for NOJD and derive flows based on these cost functions for NOJD. Then we define and investigate the noise sensitivity of the stationary points of these flows. We show that the condition number of the joint diagonalizer and uniqueness of the joint diagonalizer as measured by modulus of uniqueness (as defined in this paper) affect the sensitivity. We also investigate the effect of the number of matrices on the sensitivity. Our numerical experiments confirm the theoretical results.[1]

Key words. joint diagonalization, independent component analysis (ICA), simultaneous diagonalization, sensitivity analysis, perturbation analysis, CANDECOMP/PARAFAC, tensor decompositions

AMS subject classifications. 15A23, 15A69, 15A52, 74P20, 49Q12

DOI. 10.1137/060655997

1. Introduction and a case study.

Many interesting recent problems and paradigms in blind signal processing can be formulated as the problem of matrix joint diagonalization (JD). This problem in its simplest form can be phrased as follows: given a set of N symmetric matrices $\{C_i\}_{i=1}^N$ of dimension $n \times n$, find a nonsingular matrix B such that all BC_iB^T's are "as diagonal as possible," where B^T denotes the transpose of matrix B. Note that here diagonalization is meant in the sense of congruence. The matrix joint diagonalization problem is also referred to as simultaneous matrix diagonalization. In practice, i.e., when C_i's are constructed from empirical data, we do not expect a B to exist such that all BC_iB^T's are diagonal. Therefore, maybe a more exact name for this problem can be approximate joint diagonalization. Nevertheless, we choose to call this problem as joint diagonalization, where approximation is implicitly assumed, and we refer to the problem when exact joint diagonalization is possible as exact joint diagonalization (EJD).

Historically, the problem of matrix joint diagonalization in the signal processing community was first considered in the restricted form of orthogonal joint diagonalization (OJD) in [9], where an efficient algorithm for it was proposed. In the OJD problem the joint diagonalizer is assumed to be orthogonal. This situation can happen, for example, when one tries to blindly separate non-Gaussian sources that are spatially whitened [9]. The orthogonality assumption on B is not justified in many occasions, and one expects that by allowing more freedom in the search space "more

*Received by the editors April 3, 2006; accepted for publication (in revised form) by P. Comon August 21, 2007; published electronically September 25, 2008. This research was supported in part by the Army Research Office under ODDR&E MURI01 Program grant DAAD19-01-1-0465 to the Center for Communicating Networked Control Systems (through Boston University).

http://www.siam.org/journals/simax/30-3/65599.html

†Department of Applied Mathematics, University of Maryland, College Park, 20740 MD (bijan@umd.edu).

[1]While this paper was under review a few of its results were presented in ICASSP07 conference in Honolulu, HI [4].

diagonalization" would be possible. We refer to the problem of joint diagonalization, when B is only assumed to be nonsingular, as the problem of nonorthogonal joint diagonalization (NOJD). The focus of this paper is the NOJD problem.

Nonorthogonal joint diagonalization arises in a variety of problems. As a case study, we will see how the problem of independent component analysis (ICA) can be considered as an NOJD problem. In the problem of blind separation of nonstationary mixtures [21], one can perform NOJD on a set of correlation matrices to find the unmixing matrix. Blind separation of instantaneous mixtures using only second order statistics also results in NOJD of a set of covariance matrices [7]. Moreover, the NOJD problem is closely related to the problem of tensor decomposition and CANDECOMP/PARAFAC modeling [14, 11]. Since applications or algorithms are not the focus of this work, we will not cite numerous applications where the NOJD problem is useful. Instead, we consider the ICA problem, as a case study, to give the reader a feeling of the recurring situation, where the NOJD problem shows itself in numerous applications.

1.1. A case study: Independent component analysis (ICA). ICA [10] is one of the major paradigms in which joint diagonalization and tensorial methods have proven useful. We refer the reader to [13] for further discussion on this issue. The basic model in ICA is

$$\vec{\mathbf{x}}_{n\times 1} = A_{n\times n}\vec{\mathbf{s}}_{n\times 1}, \tag{1}$$

where $\vec{\mathbf{s}}_{n\times 1}$ is a random vector of dimension n with independent components of zero mean, and A is an $n \times n$ nonsingular matrix. We can think of $\vec{\mathbf{s}}_{n\times 1}$ to represent a source with independent components whose signals are mixed by the mixing matrix A and $\vec{\mathbf{x}}_{n\times 1}$ to represent the observed mixture. The problem is to find the matrix A or its inverse, assuming that only realizations or the moments of the random mixture $\vec{\mathbf{x}}_{n\times 1}$ are available . Obviously, we can only hope to find A up to column permutation and column scaling. The key assumption of independence of the elements of $\vec{\mathbf{s}}$ imposes some specific structure on certain matrices that can be formed from the cumulants of the observation $\vec{\mathbf{x}}$. The main theme here is that independence implies diagonality. We investigate this further. First, note that $R_{\mathbf{xx}}$, the covariance matrix of $\vec{\mathbf{x}}$, satisfies

$$R_{\mathbf{xx}} = A\Lambda_{\mathbf{ss}}A^T, \tag{2}$$

where $\Lambda_{\mathbf{ss}}$ is the (diagonal) covariance matrix of $\vec{\mathbf{s}}$. We can trace this structure in higher cumulants of $\vec{\mathbf{x}}$ as well. The kth order cumulant of a random vector $\vec{\mathbf{z}}_{n\times 1}$ is a tensor $\mathcal{C}^k_{\mathbf{z}}$ of order k and dimension $n \times \cdots \times n$. The cumulants are closely related to the moments, and they give information about the shape of the probability density function of $\vec{\mathbf{z}}_{n\times 1}$. In fact, the second order cumulant tensor is the covariance matrix. Each element of $\mathcal{C}^k_{\mathbf{z}}$ can be indexed by k indices $i_1, \ldots, i_k$, where $1 \le i_1, \ldots, i_k \le n$. If we fix all but two indices and vary the remaining two indices we obtain a matrix slice of the tensor. The notation $\mathcal{C}^k_{\mathbf{z}}(i_1, i_2, \ldots, i_{k-2}, :, :)$ represents such a matrix that is found by fixing all but the last two indices. An important fact is that, if $\vec{\mathbf{z}}_{n\times 1}$ is of independent components, then its cumulant tensors of any order are diagonal. Since $\vec{\mathbf{s}}_{n\times 1}$ is of independent components, its cumulant tensors are diagonal, i.e., only the elements $\mathcal{C}^k_{\mathbf{s}}(i, \ldots, i)$ can be nonzero. Based on the multilinear property of cumulants we can show that for $k \ge 3$:

$$\mathcal{C}^k_{\mathbf{x}}(i_1, i_2, \ldots, i_{k-2}, :, :) = A\Lambda_{i_1 i_2 \ldots i_{k-2}}A^T, \tag{3}$$

where $\Lambda_{i_1 i_2 \dots i_{k-2}}$ is a diagonal matrix that depends on elements of A and $\mathcal{C}_{\mathbf{s}}^k(i,\dots,i)$'s, the auto-cumulants of $\vec{\mathbf{s}}$, as

$$[\Lambda_{i_1 i_2 \dots i_{k-2}}]_{ii} = a_{i_1 i} a_{i_2 i} \dots a_{i_{k-2} i} \mathcal{C}_{\mathbf{s}}^k(i,\dots,i), \quad 1 \le i \le n. \tag{4}$$

Note that (2) is also of this form except that the diagonal matrix $\Lambda_{\mathbf{ss}}$ does not depend on A. There is a profound difference between cumulant matrix slices of order higher than two and the covariance matrix of $\vec{\mathbf{x}}_{n\times 1}$, in that the latter is always positive definite whereas the former need not be of any definite sign, and their signs depend both on the signs of the $\mathcal{C}_{\mathbf{s}}^k(i,\dots,i)$'s as well as the elements of A. From (2) and (3) one can see how NOJD and ICA are related: in order to find A^{-1}, search for a nonsingular matrix B that jointly diagonalizes all the cumulant matrix slices, including the covariance matrix. In section 2.2 we show that under certain conditions, which are basically the uniqueness conditions for the EJD problem, A can be found (up to the inherent indeterminacies) from the NOJD of the cumulant slices. The interesting point here is that restoration of diagonality can be equivalent to restoration of independence, and in this process we do not need to know much about the source $\vec{\mathbf{s}}_{n\times 1}$ or its statistical distribution.

1.2. Scope and organization of the paper. In [29, 3, 6, 28, 25, 1, 20] and many other works, different algorithms have been proposed to find the nonorthogonal joint diagonalizer of a given set of matrices. Although one might think of other ideas, the NOJD problem has been considered as a minimization problem whose solution gives the joint diagonalizer. There are not so many cost functions known that can be used for this purpose. Given a set of matrices

$$C_i \approx A \Lambda_i A^T, \quad 1 \le i \le N, \tag{5}$$

where Λ_i's are diagonal, the hope of NOJD is that if a B if found such that all BC_iB^T's are "as diagonal as possible," then B is close to A^{-1} up to permutation and diagonal scaling. Therefore, the accuracy or usefulness of a NOJD algorithm depends on the actual algorithm and on the cost function used, in the sense that how its minimizers differ from A^{-1} when we have (5) instead of an equality. The focus of this work is on what factors affect the sensitivity of the NOJD cost functions. Using a perturbation analysis for the stationary points of certain minimization flows, we will show that this sensitivity is closely related to the uniqueness properties of the corresponding exact joint diagonalization problem. Also, not unexpectedly, we show that if norm of A^{-1} is large, then again the NOJD will be sensitive. Note that this can happen if the norm of A is small or if A is ill-conditioned. One of our main motivations in considering the sensitivity issue is to investigate the effect of the number of matrices included in the NOJD process. Inclusion of more matrices cannot only help to reduce the harm of noise by an averaging effect but also by reducing the sensitivity through improvement of measures of uniqueness defined in section 2.

The organization of this paper is as follows: in section 2 we investigate the uniqueness conditions for the problem of exact joint diagonalization. We also use this result to derive the well-known identifiability conditions for the ICA problem [10]. In section 3 we introduce some of the known cost functions for NOJD and derive the corresponding flows whose stationary points characterize the joint diagonalizers. In section 4 we perform a perturbation analysis on the stationary points of the introduced flows in order to find the sensitivity properties. We also elaborate on the effect of the number of matrices in the NOJD process. Numerical simulations in section 5 confirm the derived results.

1.3. Notations. Throughout the paper all variables are real valued unless otherwise stated. Boldface small letters denote random variables. A and B both are $n \times n$ nonsingular matrices unless otherwise stated. If X is a matrix, then x_{ij} or X_{ij} or $[X]_{ij}$ denotes its entry at position (i, j). $\|X\|_F$ and $\|X\|_2$ denote the Frobenius norm and the 2-norm of the matrix X, respectively. X^T denotes the transpose of X, and X^{-T} denotes the transpose of the inverse of X. $\mathrm{tr}(X)$ is the trace of the square matrix X. $\mathrm{cond}(A)$ is the 2-norm based condition number of the matrix A. For a square matrix, $\mathrm{diag}(X)$ is the diagonal part of X, i.e., a diagonal matrix whose diagonal is equal to the diagonal of X. I or $I_{n\times n}$ denotes the $n \times n$ identity matrix. Unless otherwise stated, letters D and Π denote a nonsingular diagonal matrix and a permutation matrix, respectively. For a vector x, $\mathrm{diag}(x)$ is a diagonal matrix with diagonal x. Λ_i is a diagonal matrix and we denote the kth diagonal element of Λ_i by λ_{ik}. $\|x\|$ is the 2-norm of the vector x. We also define $X^\circ = X - \mathrm{diag}(X)$. $\mathrm{GL}(n)$ and $\mathrm{SO}(n)$ denote the Lie groups of nonsingular $n \times n$ matrices and orthogonal $n \times n$ matrices with $+1$ determinant, respectively. T_pM denotes the tangent space of the manifold M at point p on the manifold. Notation $X \leftarrow Y$ means that "the new value of X is Y."

2. Uniqueness conditions for exact joint diagonalization (EJD). Consider matrices

$$C_i = A\Lambda_i A^T, \quad 1 \le i \le N, \tag{6}$$

where Λ_i's are diagonal matrices, i.e., $\Lambda_i = \mathrm{diag}([\lambda_{i1}, \ldots, \lambda_{in}])$. One interesting problem is that given only $\{C_i\}_{i=1}^N$, find A. We call this problem the exact joint diagonalization (EJD) problem. Note that with only the information that Λ_i's are diagonal, A can be determined only up to permutation and diagonal scaling, i.e., if A is a solution then $AD\Pi$ is also a solution, for any D and Π. We say that the EJD has a unique[2] solution if the permutation and diagonal scaling are the only ambiguities in finding A. If the EJD has a unique solution, then finding A is equivalent to finding a $B \in \mathrm{GL}(n)$ such that all BC_iB^T's are diagonal, hence the name "joint diagonalization."

The issue of uniqueness in the EJD problem can be considered as a special case of uniqueness in the CANDECOMP/PARAFAC model, which has been addressed in [16]. In order to quantify the uniqueness property, which as will be seen in section 4 is closely related to the sensitivity issue of the NOJD problem, we rephrase the necessary and sufficient conditions for uniqueness differently from the related literature.

DEFINITION 1. *For the set of diagonal matrices $\{\Lambda_i\}_{i=1}^N$, let*

$$\rho_{kl} = \frac{\displaystyle\sum_{i=1}^{N} \lambda_{ik}\lambda_{il}}{\left(\displaystyle\sum_{i=1}^{N} \lambda_{il}^2\right)^{\frac{1}{2}} \left(\displaystyle\sum_{i=1}^{N} \lambda_{ik}^2\right)^{\frac{1}{2}}}, \quad 1 \le k \ne l \le N, \tag{7}$$

with the convention that $\rho_{kl} = 1$ if $\lambda_{ik} = 0$ for some k and all i. Let ρ be equal to one of the ρ_{kl}'s that have the maximum absolute value among all. The modulus of uniqueness for this set is defined as $|\rho|$.

Note that $|\rho| \le 1$ and $|\rho| = 1$ if and only if at least two columns of the matrix $[\Lambda]_{ij} = \lambda_{ij}$ are collinear, i.e., if there is a real number K and integers p and q such that

[2] In some works this is referred to as "essential uniqueness."

$\lambda_{ip} = K\lambda_{iq}$, for $1 \leq i \leq N$. $|\rho|$ measures the maximum degree of collinearity between any two columns of the matrix $[\Lambda]_{ij} = \lambda_{ij}$. This measure to quantify collinearity may seem to be chosen arbitrarily, but as will be seen later it shows itself naturally in the analysis of certain cost functions for NOJD. Another measure, which also naturally appears in the analysis of the log-likelihood based cost (see section 3.2.3) function is given in the following definition.

DEFINITION 2. *For the set of positive definite diagonal matrices $\{\Lambda_i\}_{i=1}^N$, let*

$$\mu_{kl} = \frac{1}{N^2}\left(\sum_{i=1}^{N}\frac{\lambda_{ik}}{\lambda_{il}}\right)\left(\sum_{i=1}^{N}\frac{\lambda_{il}}{\lambda_{ik}}\right), \quad 1 \leq k \neq l \leq N. \tag{8}$$

Let μ be the minimum value of μ_{kl}'s. The modulus of uniqueness of second type for this set is defined as μ.

Note that $\mu \geq 1$ with equality if and only if $|\rho| = 1$. μ also measures the maximum collinearity between the columns of Λ, with the assumption that Λ_i's are positive definite.

If $N = 1$, then $|\rho| = 1$ and the diagonalizer is not unique. For $N > 1$, the modulus of uniqueness also captures the uniqueness property in an exact sense.

THEOREM 1. *Let C_i's satisfy* (6). *The necessary and sufficient condition to have unique nonorthogonal joint diagonalizer is that $|\rho| < 1$.*

Proof. First we consider the case $n = 2$. If $|\rho| = 1$, then either (a) there is a real number K such that $\lambda_{i2} = K\lambda_{i1}$ for all $1 \leq i \leq N$, or (b) $\lambda_{i1} = 0$ for all $1 \leq i \leq N$ and $\lambda_{i2} \neq 0$ for some i, or (c) $\lambda_{i1} = \lambda_{i2} = 0$ for all i, which is a trivial situation. In case of (a) we have:

$$C_i = \lambda_{i1} A \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & \sqrt{|K|} \end{bmatrix}}_{D_K} \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{|K|} \end{bmatrix} A^T. \tag{9}$$

We have denoted the diagonal matrix that includes $\sqrt{|K|}$ as D_K. Let us first assume that $K \neq 0$. Now, if $\rho = +1$, then let $B = Q_{+1}D_K^{-1}A^{-1}$, where

$$Q_{+1} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{10}$$

This B diagonalizes every C_i for all θ. If $\rho = -1$, then let $B = Q_{-1}D_K^{-1}A^{-1}$, where

$$Q_{-1} = \begin{bmatrix} \cosh\theta & \sinh\theta \\ \sinh\theta & \cosh\theta \end{bmatrix}. \tag{11}$$

This B diagonalizes every C_i for all θ. If $K = 0$ (and hence $\rho = 1$), then let $B = Q_1A^{-1}$, where

$$Q_1 = \begin{bmatrix} 1 & \theta \\ 0 & 1 \end{bmatrix}. \tag{12}$$

This B diagonalizes every C_i for all θ. Also, in case of (b), $B = Q_1^TA^{-1}$ diagonalizes every C_i for all θ. Therefore, for $|\rho| = 1$ the nonorthogonal joint diagonalizer is not unique. For $n > 2$, $|\rho| = 1$ means that the situation described for $n = 2$ happens between two diagonal elements, in the same positions within Λ_i's; and we can apply

the previous argument to those elements. Hence, for $n > 2$ also $|\rho| = 1$ implies nonuniqueness of the joint diagonalizer. To see the necessary part, first note that existence of more than one exact joint diagonalizer means that there exists a C which differs from a permuted diagonal matrix such that the matrices $D_i = C\Lambda_i C^T$ are diagonal. For the moment, assume that one of the Λ_i's, say Λ_1, is nonsingular. Then $D_i D_1^{-1} = C\Lambda_i\Lambda_1^{-1}C^{-1}$ for $1 < i \leq N$. These are the eigendecompositions of diagonal matrices $\Lambda_i\Lambda_1^{-1}$'s, for $1 < i \leq N$. Nonuniqueness of C happens only when there are two integers $1 \leq k \neq l \leq n$, such that $\frac{\lambda_{ik}}{\lambda_{1k}} = \frac{\lambda_{il}}{\lambda_{1l}}$, $1 < i \leq N$. This means that $|\rho| = 1$. If C is not unique and all Λ_i's are singular, then two cases can happen. In the first case, all Λ_i's have one zero diagonal element at a common position, i.e., there exists an integer $1 \leq k \leq n$ such that for all $1 \leq i \leq N$ we have $\lambda_{ik} = 0$, which implies that $\rho = 1$. If the first case is not true, then there exists a linear combination of Λ_i's like Λ_0, which is nonsingular, and $D_0 = C\Lambda_0 C^T$ is diagonal; then we are back to the nonsingular case. This completes the proof. □

This result and more general ones have been referred to in [23] using the concept of Kruskal's rank.

2.1. On the minimum number of matrices needed for EJD. Let A be an orthogonal matrix. Then the equations in (6) are the eigendecompositions of C_i's. If C_1 or, equivalently, Λ_1 has distinct eigenvalues, then A can be found from eigendecomposition of C_1, uniquely up to permutations. If Λ_1 has only two equal diagonal elements at positions k and l, and if we can find another Λ_i with distinct values at those positions, then A can again be found uniquely from eigendecompositions of C_1 and C_i. Therefore, if for each pair of k and l we can find an i for which $\lambda_{il} \neq \lambda_{ik}$, then A can be determined uniquely. As a result, in the generic case orthogonal joint diagonalization is in fact a one-matrix problem and the inclusion of more matrices can be justified by the presence of noise. The uniqueness properties of OJD as well as its sensitivity analysis have been addressed in [8].

There is a huge difference between the uniqueness properties of the orthogonal and nonorthogonal joint diagonalization problems. From the proof of Theorem 1 it should be evident that $N = 1$ matrix is not enough to find a unique nonorthogonal (joint) diagonalizer. NOJD allows more degrees of freedom in finding the diagonalizer. Let us count the degrees of freedom in both sides of the equations in (6). Recall that a symmetric $n \times n$ matrix has $\frac{n(n+1)}{2}$ degrees of freedom, and A has $n^2 - n$ degrees of freedom (as far as the NOJD problem is concerned). Hence, the left-hand side of (6) has total $N\frac{n(n+1)}{2}$ degrees of freedom, and its right-hand side has $n^2 - n + Nn$ degrees of freedom. Equating the degrees of freedom from both sides and solving for N gives $N = 2$. Therefore, the minimum number of matrices to give enough equations to find a unique nonorthogonal joint diagonalizer is $N = 2$; and hence, the NOJD problem, in the generic case, is a two-matrix problem. For two arbitrary and generic matrices $\{C_1, C_2\}$, whether the equations in (6) yield a real valued solution for $\{A, \Lambda_1, \Lambda_2\}$ depends on the matrices.[3] It is well known that if one of the two matrices is positive definite, then they admit a (real) exact jointly diagonalizer [15, pp. 461–462]. As we will show in section 4.4.1, for only two matrices, if their dimension is moderately large ($n > 20$, for example), then the modulus of uniqueness is close to unity. This in turn, as will be shown in section 4, means that the NOJD of the two matrices is an

[3]Assuming C_1 is invertible (which is true for a generic matrix), in order for (6) to hold, we should have that $C_2 C_1^{-1} = A\Lambda_2\Lambda_1^{-1}A^{-1}$, which is an eigendecomposition. Again in a generic case, this would give a unique and (in general) complex valued $\{A, \Lambda_1, \Lambda_2\}$.

ill-conditioned problem; and hence, it is better to include more matrices in the NOJD process.

2.2. Identifiability of the ICA problem. Now we would like to apply the previous theorem to the case of the ICA problem. It is obvious that if we can find two cumulant matrix slices of $\vec{\mathbf{x}}_{n\times 1}$ for which $|\rho|$ is not unity, then the matrix A in (1) can be found uniquely. From (3) and (4), one can show that for the set $\{\mathcal{C}^k_{\mathbf{x}}(i_1, i_2, \dots, i_{k-2}, :, :)\}_{1\le i_1,\dots,i_{k-2}\le n}$ with $k > 2$ we have $|\rho| \neq 1$, if and only if none of $\mathcal{C}^k_{\mathbf{s}}(i, \dots, i)$'s are zero. To see this, first note that if $\mathcal{C}^k_{\mathbf{s}}(i, \dots, i) = 0$ for some i, then $|\rho| = 1$. Now assume that none of $\mathcal{C}^k_{\mathbf{s}}(i, \dots, i)$'s are zero, and $|\rho| = 1$. Since $|\rho| = 1$, there are two columns of A like j and l and a real number K such that

$$a_{i_1 j} a_{i_2 j} \dots a_{i_{k-2} j} \mathcal{C}^k_{\mathbf{s}}(j, \dots, j) = K a_{i_1 l} a_{i_2 l} \dots a_{i_{k-2} l} \mathcal{C}^k_{\mathbf{s}}(l, \dots, l) \tag{13}$$

for all $1 \le i_1, ..., i_{k-2} \le n$. Because none of the $\mathcal{C}^k_{\mathbf{s}}(i, ..., i)$'s are zero, and since there is at least one nonzero element like a_{pj} in the j^{th} column of A, by setting $i_2 = \dots = i_{k-2} = p$ we have that there is another real number K' such that

$$a_{i_1 j} = K' a_{i_1 l} \tag{14}$$

for all $1 \le i_1 \le n$. This contradicts the invertibility of A. Hence, with an invertible A, $|\rho|$ cannot be unity unless at least one of $\mathcal{C}^k_{\mathbf{s}}(i, \dots, i)$'s is zero.

Now assume that the covariance matrix of $\vec{\mathbf{s}}_{n\times 1}$ is nonsingular, i.e., there is no source component with zero variance. Then by inclusion of the covariance matrix of $\vec{\mathbf{x}}_{n\times 1}$ in the above set we can weaken the uniqueness condition, i.e., for $\{R_{\mathbf{xx}}, \mathcal{C}^k_{\mathbf{x}}(i_1, i_2, \dots, i_{k-2}, :, :)\}_{1\le i_1,\dots,i_{k-2}\le n}$ with $k > 2$, we have $|\rho| \neq 1$ if and only if at most one of $\mathcal{C}^k_{\mathbf{s}}(i, \dots, i)$'s is zero. Therefore, if we start with the covariance matrix of $\vec{\mathbf{x}}_{n\times 1}$ and then include its third order cumulant slices, and if at most one of the skewness' $\mathcal{C}^3_{\mathbf{s}}(i, \dots, i)$ is zero, then A can be determined uniquely. If at least two $\mathcal{C}^3_{\mathbf{s}}(i, \dots, i)$'s are zero, then we can go to the cumulants of higher orders and check the same condition. Note that this process fails if and only if there are at least two source elements s_p and s_q for which $\mathcal{C}^k_{\mathbf{s}}(p, \dots, p) = \mathcal{C}^k_{\mathbf{s}}(q, \dots, q) = 0$ for all $k \ge 3$. It is well known that such random variables have Gaussian distribution. As a result, exact nonorthogonal joint diagonalization of the set of all cumulant matrix slices of $\vec{\mathbf{x}}_{n\times 1}$ gives A uniquely, unless $\vec{\mathbf{s}}_{n\times 1}$ has at least two Gaussian components. To summarize we state this theorem (cf. Corollary 13 in [10]).

THEOREM 2 (identifiability of ICA:EJD formulation). *Consider the model* (1). *About* $\vec{\mathbf{s}}_{n\times 1}$*, assume that its covariance matrix is nonsingular, its* k*th order cumulants (for some* $k > 2$*) exist, and at most one of them is zero. Then exact joint diagonalization of the set* $\{R_{\mathbf{xx}}, \mathcal{C}^k_{\mathbf{x}}(i_1, i_2, \dots, i_{k-2}, :, :)\}_{1\le i_1,\dots,i_{k-2}\le n}$ *results in finding* A *up to column permutation and scaling. For a source vector with finite cumulants of all orders, this process fails to identify* A *if only if the source vector has more than one Gaussian component.*

This result suggests that EJD can be used as a basis to define a contrast function [10] for ICA. Note that this identifiability condition is derived solely based on the algebraic structure of the ICA model and that we have not used the Skitovich–Darmois theorem [10, 18]. OJD or NOJD of cumulant matrix slices of order three, four, or even higher have been suggested in many works, e.g., [9, 28, 29, 19, 13, 2]. The OJD scenario arises when one assumes that the mixture is already uncorrelated or whitened.

3. Cost functions for joint diagonalization. The joint diagonalization problem has been posed, in the literature, mostly as an optimization problem [9, 28, 25, 20]. We mention that in [28, 25] the joint diagonalization problem has been addressed with a different formulation than ours. As mentioned before, generically, in the OJD problem one matrix ($N = 1$), and in the NOJD problem two matrices ($N = 2$) are enough to find a unique joint diagonalizer. However, it is believed that inclusion of more matrices is useful in making the solution less vulnerable to noise. Therefore, the proposed cost functions for joint diagonalization are designed to mitigate the effect of noise via averaging.

3.1. A cost function for orthogonal joint diagonalization. The OJD problem was introduced earlier than the NOJD problem. In [9] a natural cost function together with an efficient algorithm for OJD was introduced. The cost function $J_1 : \mathrm{SO}(n) \to \mathbb{R}$ for OJD, introduced in [9], is

$$J_1(\Theta) = \sum_{i=1}^{n} \left\| \Theta C_i \Theta^T - \mathrm{diag}(\Theta C_i \Theta^T) \right\|_F^2, \tag{15}$$

where $\{C_i\}_{i=1}^N$ is the set of symmetric matrices to be diagonalized. If Θ minimizes J_1, then we call Θ an orthogonal joint diagonalizer of $\{C_i\}_{i=1}^N$. Note that, since $\mathrm{SO}(n)$ is a compact manifold, a priori we know that a minimizer exists for J_1. Whether, generically, this cost function has only global minimum on $\mathrm{SO}(n)$, and whether the minimizers are unique up to permutation are not known.

3.2. Cost functions for nonorthogonal joint diagonalization. Introducing a cost function for NOJD has been a challenge. First, note that a simple extension of J_1 from $\mathrm{SO}(n)$ to $\mathrm{GL}(n)$ is not effective. We remind that the NOJD problem in the exact case is a scale-invariant problem, i.e., if $B \in \mathrm{GL}(n)$ is an exact joint diagonalizer for a set of matrices, then DB also should be a joint diagonalizer for any nonsingular diagonal D. We expect or require this to be true for the case of NOJD as well, i.e., if B is a nonorthogonal joint diagonalizer for a set of matrices, we expect DB also to be a nonorthogonal joint diagonalizer for that set. However, in general $J_1(DB) \neq J_1(B)$. In fact, we can reduce $J_1(B)$ just by reducing the norm of B, and $J_1(B)$ has a global infimum at $B = 0$.

3.2.1. A nonholonomic flow for NOJD based on J_1. For the derivations in this subsection we refer the reader to [2]. We also refer the reader to [17] for more comprehensive treatment of gradient flows for optimization on manifolds. On the Lie group of nonsingular matrices, we can define a right-invariant Riemannian metric[4] that matches the group structure as

$$\begin{aligned} \langle ., . \rangle_B &: T_B GL(n) \times T_B GL(n) \to \mathbb{R} \\ \langle \xi_1, \xi_2 \rangle_B &= \mathrm{tr}((\xi_1 B^{-1})^T \xi_2 B^{-1}), \end{aligned} \tag{16}$$

where $T_B\mathrm{GL}(n)$ is the tangent space to $\mathrm{GL}(n)$ at B. In general, a tangent vector ξ at a point B on $\mathrm{GL}(n)$ (and any Lie group) can be written as $\xi = \zeta B$ where ζ belongs to

[4]The significance of the right-invariant metric is that it matches the invariance property of the NOJD problem, which as mentioned is that the joint diagonalizer does not change by left multiplication by nonsingular diagonal matrices. A discretization of a right-invariant flow such as $\frac{dB}{ds} = \Omega B$ has the form $B_{k+1} = (I + \Omega_k) B_k$.

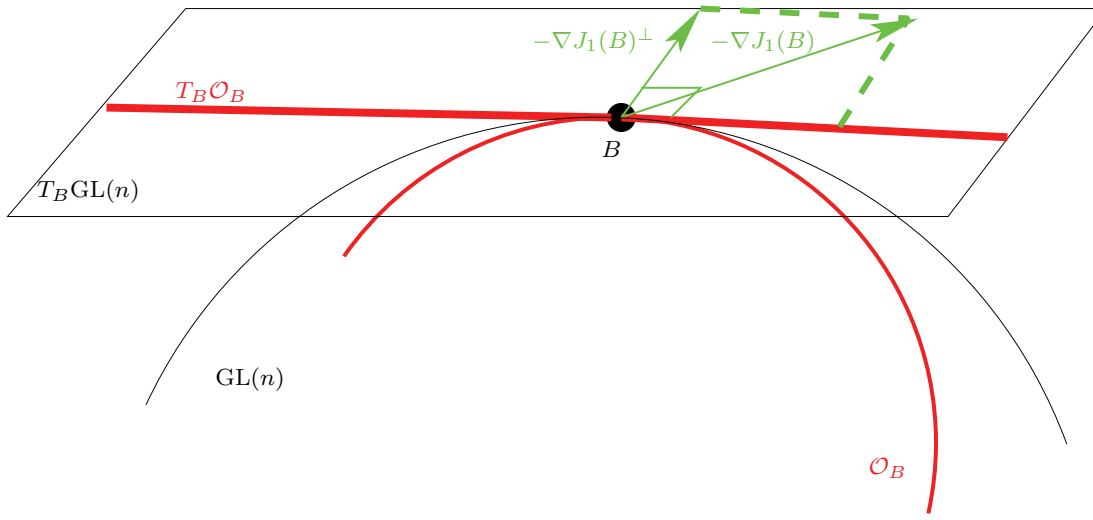


FIG. 1. *The group of nonsingular diagonal matrices acts on the manifold $GL(n)$ at B via left multiplication. $\mathcal{O}_B$ is the orbit of this action. The linearization of this orbit (i.e., the tangent space to it) at B is $T_B\mathcal{O}_B \subset T_B GL(n)$. This figure shows how $-\nabla J_1(B)$ should be projected onto the orthogonal complement of $T_B\mathcal{O}_B$ in order to have a flow for NOJD based on J_1 which is not a scale-invariant cost function for NOJD.*

the tangent space at the identity. Also, the tangent spaces at B and the identity are isometric in the previous metric. Let $s \mapsto \tilde{B}(s)$ be any smooth curve with $\tilde{B}(0) = B$. With respect to the Riemannian metric in (16), the gradient of $J_1 : \mathrm{GL}(n) \to \mathbb{R}$ is defined as a vector field ∇J_1 that satisfies

$$\dot{J}_1 = \langle \nabla J_1, \dot{B} \rangle_B, \tag{17}$$

where $\dot{J}_1 = \frac{dJ_1(\tilde{B}(s))}{ds}\big|_{s=0}$ and $\dot{B} = \frac{d\tilde{B}(s)}{ds}\big|_{s=0}$. From this, it is easy to verify that up to a scalar factor

$$\nabla J_1(B) = \Omega_1 B, \tag{18}$$

where

$$\Omega_1 = \sum_{i=1}^{N} (BC_iB^T)^{\circ} BC_iB^T. \tag{19}$$

We can show that the stationary points of J_1, i.e., values of B for which $\nabla J_1(B) = 0$ and hence $\Omega_1 = 0$, satisfy $BC_iB^T = \mathrm{diag}(BC_iB^T)$. Therefore, if C_i's do not have an exact joint diagonalizer, then J_1 will have no stationary points on $\mathrm{GL}(n)$. A gradient flow for minimization of J_1 has the form $\frac{dB}{ds} = -\nabla J_1(B) = -\Omega_1 B$. As we mentioned before, the problem with minimizing J_1 as a cost function for NOJD is that it can be reduced by diagonal matrices. At each point $B \in \mathrm{GL}(n)$, we can project the gradient of J_1 (or more accurately the negative of the gradient) to directions that do not correspond to diagonal scaling. The group of nonsingular diagonal matrices of dimension n can act on the group $\mathrm{GL}(n)$ via left multiplication. At B the orbit of this action is simply $\mathcal{O}_B = \{DB | D = \text{nonsigngular and diagonal}\}$ and it is in fact a

submanifold which we would like our NOJD flow to avoid. The linearization or the tangent space to the orbit at B is $T_B\mathcal{O}_B = \{DB|D = \text{diagonal}\}$, which is a linear subspace of $T_B\mathrm{GL}(n)$. The orthogonal complement of $T_B\mathcal{O}_B$ in the tangent space $T_B\mathrm{GL}(n)$, with respect to the defined Riemannian metric, is $(T_B\mathcal{O}_B)^\perp = \{\Xi B|\Xi \in \mathbb{R}^{n\times n},\ \mathrm{diag}(\Xi) = 0\}$. Therefore, the projection of ∇J_1 onto $(T_B\mathcal{O}_B)^\perp$ is $\nabla J_1^\perp = \Omega_1^\circ B$. Figure 1 shows the process of constraining the negative of the gradient to directions along $(T_B\mathcal{O}_B)^\perp$, at each point B. The corresponding nonholonomic[5] flow for NOJD is

$$\frac{dB}{ds} = -\nabla J_1^\perp = -\Omega_1^\circ B. \tag{20}$$

The stationary points or equilibria of this flow are defined by $\Omega_1^\circ = 0$, or

$$\sum_{i=1}^{N} \left((BC_iB^T)^\circ BC_iB^T\right)^\circ = 0. \tag{21}$$

Hence, if a nonorthogonal joint diagonalizer of $\{C_i\}_{i=1}^N$ based on the above nonholonomic flow exists, then it should satisfy (21). In [3, 29] and many other works, minimization schemes for J_1 are proposed, which try to find the stationary points in (21).

3.2.2. A Frobenius norm scale-invariant cost function. Note that $J_1 : \mathrm{SO}(n) \to \mathbb{R}$ in (15) can also be written as

$$J_1(\Theta) = \sum_{i=1}^{n} \left\|C_i - \Theta^{-1}\mathrm{diag}(\Theta C_i\Theta^T)\Theta^{-T}\right\|_F^2. \tag{22}$$

Let $J_2 : \mathrm{GL(n)} \to \mathbb{R}$ be the extension of this form of J_1 to $\mathrm{GL}(n)$ defined by

$$J_2(B) = \sum_{i=1}^{n} \left\|C_i - B^{-1}\mathrm{diag}(BC_iB^T)B^{-T}\right\|_F^2. \tag{23}$$

Then it is easy to check that $J_2(\Pi DB) = J_2(B)$ for any nonsingular diagonal D and permutation Π. Therefore, J_2 is a scale and permutation invariant cost function for NOJD. Note that J_1 and J_2 are scaled versions of each other in the sense that

$$\frac{J_1(B)}{n^2\|B\|_2^4} \le J_2(B) \le n^2\|B^{-1}\|_2^4 J_1(B). \tag{24}$$

Also, note that we can reduce J_2 without changing the norm of B. This means that reducing J_2, if the norm of B is not changed, can result in reduction of the upper bound of J_1. This cost function has been introduced in [5, 3].

3.2.3. Log-likelihood function for NOJD. In [20], another cost function for NOJD of a set of positive definite matrices $\{C_i\}_{i=1}^N$ has been introduced. This cost function has the form

$$J_3(B) = \sum_{i=1}^{N} \log\left(\frac{\det \mathrm{diag}(BC_iB^T)}{\det BC_iB^T}\right). \tag{25}$$

[5] A nonholonomic flow is a flow whose velocity vector field is constrained by nonintegrable constraints.

A matrix $B \in \mathrm{GL}(n)$ that minimizes J_3 is the joint diagonalizer of $\{C_i\}_{i=1}^N$. It can be shown that $J_3(B) \geq 0$, and the equality holds if and only if all BC_iB^T's are diagonal. It is easy to check that J_3 is scale and permutation invariant, i.e., $J_3(\Pi DB) = J_3(B)$, for any Π and D. The specific form of this cost function is imposed by the log-likelihood function of correlation matrices of Gaussian nonstationary sources [20]. Let us consider the same right-invariant Riemannian metric as in section 3.2.1. Using the well-known identity $\frac{\partial}{\partial B} \log\det B = (B^T)^{-1}$ we can show (see (17) also) that

$$\dot{J}_3 = 2\sum_{i=1}^{N} \mathrm{tr}\Big((\dot{B}B^{-1})^T \big((\mathrm{diag}(BC_iB^T))^{-1} BC_iB^T - I \big) \Big). \tag{26}$$

As a result, with respect to the above Riemannian metric, the gradient vector field of J_3 up to a scalar factor is

$$\nabla J_3(B) = \frac{1}{N}\sum_{i=1}^{N} \big(\mathrm{diag}((BC_iB^T))^{-1} BC_iB^T - I \big) B := \Omega_3 B. \tag{27}$$

It is interesting to note that $\mathrm{diag}(\Omega_3) = 0$ (cf. (19) and (20)). A gradient flow for NOJD based on minimization of J_3 is $\frac{dB}{ds} = -\Omega_3 B$. The stationary points for this flow are characterized by $\Omega_3 = 0$; and if B is a joint diagonalizer, then it should satisfy

$$\frac{1}{N}\sum_{i=1}^{N} BC_iB^T \big(\mathrm{diag}(BC_iB^T) \big)^{-1} = I. \tag{28}$$

4. Sensitivity analysis. An interesting question to ask is: "which set of matrices are hard to be jointly diagonalized?" In other words, which factors affect the condition or sensitivity of the joint diagonalization problem? Consider the matrices $C_i = A\Lambda_i A^T,\ 1 \leq i \leq N$, where Λ_i's are diagonal. Obviously, $\{C_i\}_{i=1}^N$ have a joint diagonalizer $B = A^{-1}$. Note that, here equality is understood up to permutation and diagonal scaling. Now, we add noise to the matrices as

$$C_i = A\Lambda_i A^T + tN_i, \quad t \in [-\delta, \delta], \delta > 0, \tag{29}$$

where $\{N_i\}_{i=1}^N$ are symmetric error or noise matrices, and t shows the noise gain or contribution. The joint diagonalizer of this noisy set will deviate from A^{-1} as t deviates from zero. If the sensitivity is high, then the deviation from A^{-1} will be large. In this case, we say that the NOJD problem is very sensitive or ill-conditioned. Note that the true goal of NOJD is to find A and not just diagonalizing the matrices $\{C_i\}_{i=1}^N$. It is in this context that the sensitivity of the problem is defined. If the modulus of uniqueness for $\{\Lambda_i\}_{i=1}^N$ is unity, then $\{C_i\}_{i=1}^N$ has already infinite sensitivity; since the joint diagonalizer can change even in absence of noise. Hence, one should expect the sensitivity for joint diagonalization to be closely related to the issue of uniqueness. To quantify this relation, we will perform a perturbation analysis of the stationary points of the NOJD cost functions or flows defined in section 3. The results for different cost functions are very much similar.

4.1. Sensitivity analysis for NOJD based on J_1. The J_1-based nonorthogonal joint diagonalizer of $\{C_i\}_{i=1}^N$, B is defined by (21). As t deviates from zero in (29), $B(t)$, the joint diagonalizer, varies smoothly. For small enough δ, from the implicit function theorem and basic properties of Lie groups, we have

$$B(t) = (I + t\Delta)A^{-1} + o(t), \quad t \in [-\delta, \delta], \tag{30}$$

where $\Delta \in \mathbb{R}^{n\times n}$ with $\mathrm{diag}(\Delta) = 0$ and $\frac{\|o(t)\|}{t} \to 0$ as $t \to 0$. The restriction $\mathrm{diag}(\Delta) = 0$ matches the structure of the nonholonomic flow for NOJD derived in section 3.2.1. Note that the norm of Δ measures the sensitivity of the NOJD problem to noise. Our goal is to calculate Δ. Using $B(0) = A^{-1}$ and $\frac{dB}{dt}(0) = \Delta A^{-1}$, and after plugging (29) into (21), and then differentiating with respect to t, we can easily verify that

$$\sum_{i=1}^{N} \big(\Delta\Lambda_i + \Lambda_i\Delta^T\big)\Lambda_i = -\sum_{i=1}^{N} \big(A^{-1}N_i(A^{-1})^T\big)^{\circ}\Lambda_i. \tag{31}$$

The right-hand side of the above equation manifests the possible noise amplification that can happen due to large $\|A^{-1}\|$, i.e., when A is small in norm or more importantly when A is ill-conditioned. Equation (31) is a linear equation in terms of Δ. Let us define

$$\mathcal{T} = \sum_{i=1}^{N} (A^{-1}N_i(A^{-1})^T)^{\circ}\Lambda_i. \tag{32}$$

Now, it is easy to check that the two entries Δ_{kl} and Δ_{lk} decouple from the rest of the entries of Δ and we have

$$\begin{bmatrix} \sum_{i=1}^{N}\lambda_{il}^2 & \sum_{i=1}^{N}\lambda_{ik}\lambda_{il} \\ \sum_{i=1}^{N}\lambda_{ik}\lambda_{il} & \sum_{i=1}^{N}\lambda_{ik}^2 \end{bmatrix} \begin{bmatrix} \Delta_{kl} \\ \Delta_{lk} \end{bmatrix} = -\begin{bmatrix} \mathcal{T}_{kl} \\ \mathcal{T}_{lk} \end{bmatrix},\ 1 \le k < l \le n. \tag{33}$$

Recall the definition of ρ_{kl} (Definition 1). Also, let

$$\gamma_{kl} = \left(\sum_{i=1}^{N}\lambda_{ik}^2\right)^{\frac{1}{2}}\left(\sum_{i=1}^{N}\lambda_{il}^2\right)^{\frac{1}{2}},\quad \eta_{kl} = \frac{\left(\sum_{i=1}^{N}\lambda_{ik}^2\right)^{\frac{1}{2}}}{\left(\sum_{i=1}^{N}\lambda_{il}^2\right)^{\frac{1}{2}}}. \tag{34}$$

We denote the coefficients matrix in (33) by M_{kl},

$$M_{kl} = \gamma_{kl}\begin{bmatrix} \eta_{kl}^{-1} & \rho_{kl} \\ \rho_{kl} & \eta_{kl} \end{bmatrix},\qquad 1 \le k < l \le n. \tag{35}$$

Then (33) is equivalent to

$$\begin{bmatrix} \Delta_{kl} \\ \Delta_{lk} \end{bmatrix} = -M_{kl}^{-1}\begin{bmatrix} \mathcal{T}_{kl} \\ \mathcal{T}_{lk} \end{bmatrix} = \frac{-1}{\gamma_{kl}(1-\rho_{kl}^2)}\begin{bmatrix} \eta_{kl} & -\rho_{kl} \\ -\rho_{kl} & \eta_{kl}^{-1} \end{bmatrix}\begin{bmatrix} \mathcal{T}_{kl} \\ \mathcal{T}_{lk} \end{bmatrix},\ \ 1 \le k < l \le n. \tag{36}$$

Note that the eigenvalues of M_{kl}^{-1} are

$$\lambda_{max}, \lambda_{min} = \frac{\eta_{kl} + \eta_{kl}^{-1} \pm \sqrt{(\eta_{kl} + \eta_{kl}^{-1})^2 - 4(1-\rho_{kl}^2)}}{2\gamma_{kl}(1-\rho_{kl}^2)}. \tag{37}$$

Also, it is easy to check that

$$\frac{\eta_{kl}+\eta_{kl}^{-1}-1}{\gamma_{kl}(1-\rho_{kl}^2)} \leq \lambda_{max} < \frac{\eta_{kl}+\eta_{kl}^{-1}}{\gamma_{kl}(1-\rho_{kl}^2)}, \tag{38}$$

and

$$\frac{1}{\gamma_{kl}(\eta_{kl}+\eta_{kl}^{-1})} \leq \lambda_{min} \leq \frac{1}{\gamma_{kl}}. \tag{39}$$

Therefore, we can also establish the bounds

$$\frac{1}{\gamma_{kl}(\eta_{kl}+\eta_{kl}^{-1})} \left\| \begin{bmatrix} \Delta_{kl} \\ \Delta_{lk} \end{bmatrix} \right\| \leq \left\| \begin{bmatrix} \Delta_{kl} \\ \Delta_{lk} \end{bmatrix} \right\| < \frac{\eta_{kl}+\eta_{kl}^{-1}}{\gamma_{kl}(1-\rho_{kl}^2)} \left\| \begin{bmatrix} \mathcal{T}_{kl} \\ \mathcal{T}_{lk} \end{bmatrix} \right\|. \tag{40}$$

Because γ_{kl} is not scale-invariant, $\gamma_{kl} \approx 0$ by itself does not imply a high sensitivity. The definitions of the parameters reveal that γ_{kl} plays more of a scaling role, whereas ρ_{kl} plays a structural role. Hence, as far as sensitivity to noise is concerned, the interesting situation (approximate singularity) happens when $|\rho_{kl}| \approx 1$. Note that as $|\rho_{kl}| \to 1$, λ_{max} and λ_{min} approach their upper and lower bounds, respectively. Moreover, in that case λ_{max} grows unboundedly and λ_{min} remains bounded. Since $\mathcal{T}$ depends on random noise, there always will be a component of $\left[\begin{smallmatrix} \mathcal{T}_{kl} \\ \mathcal{T}_{lk} \end{smallmatrix}\right]$ along the direction of the eigenvector of M_{kl}^{-1} corresponding to λ_{max}. Therefore, when $|\rho_{kl}|$ approaches unity, $\| \left[\begin{smallmatrix} \Delta_{kl} \\ \Delta_{lk} \end{smallmatrix}\right] \|$ tends towards the upper bound in (40). Hence, the upper bound is the more interesting one and it is not a loose bound in the sense that it can be achieved very closely when $|\rho_{kl}| \approx 1$.[6] One can easily check that

$$\|\Delta\|_F < \frac{\alpha}{(1-\rho^2)}\|\mathcal{T}\|_F \leq \frac{n\alpha\|A^{-1}\|_2^2}{(1-\rho^2)} \sum_{i=1}^{N} \|N_i\|_2 \|\Lambda_i\|_2, \tag{41}$$

where $\alpha = \max_{k\neq l} \frac{\eta_{kl}+\frac{1}{\eta_{kl}}}{\gamma_{kl}}$, and $|\rho|$ is the modulus of uniqueness for the set $\{\Lambda_i\}_{i=1}^N$ as defined before. Since an approximate nonuniqueness of the joint diagonalizer can happen when only one of $|\rho_{kl}|$'s is close to unity, the above bound might seem exaggerative. Again one can imagine a worse case scenario in which all $|\rho_{kl}|$'s are close to unity, and the bound would not be very loose. In summary, we have the following theorem.

THEOREM 3. *Let $C_i = A\Lambda_i A^T + tN_i, 1 \leq i \leq N$ ($t \in [-\delta, \delta]$). Let us define $B(t)$ the nonorthogonal joint diagonalizer for $\{C_i\}_{i=1}^N$ as the minimizer of J_1 under the nonholonomic flow with equilibria defined in* (21). *Then for small enough δ, the joint diagonalizer can be written as: $B(t) = (I+t\Delta)A^{-1}+o(t)$, where Δ (with $diag(\Delta) = 0$) satisfies* (36) *as well as* (41).

The bound in (41) confirms the intuition that if the joint diagonalizer is close to nonuniqueness, as measured by the modulus of uniqueness, or if it is ill-conditioned, then the sensitivity of the NOJD problem will be high. Note that our derivations suggest that there is another scenario that can result in high sensitivity, which is when A and Λ_i (i.e., γ_{kl}'s) are small in norm. Of course, this is not an interesting scenario. We could have avoided this by imposing a constraint on the norm of the noise in (29). For example, we could assume that $\|N_i\|_2 \leq \|A\|_2^2\|\Lambda_i\|_2$. This choice

[6] It is common in matrix perturbation theory to have bounds that only handle the worse cases well. For a discussion on this issue, see [22, p. 124].

makes the bound (41) such that it is unchanged if A or Λ_i's all are scaled by a scalar. Hence, one might be tempted to define $\frac{\mathrm{cond}(A)^2}{1-\rho^2}$ as the condition number for the NOJD problem based on J_1.

4.2. Sensitivity analysis for NOJD based on J_2. We can follow the same path as in the previous subsection and perform a perturbation analysis for the stationary points of J_2 in presence of noise. In [5], it is shown that the stationary points of J_2 satisfy

$$\begin{cases} \sum_{i=1}^{N} \big(\Psi_i \mathrm{diag}\big(BC_iB^T\big) - \mathrm{diag}\big(\Psi_i\big)BC_iB^T\big) = 0, \\ \Psi_i = (BB^T)^{-1}\big(BC_iB^T - \mathrm{diag}(BC_iB^T)\big)(BB^T)^{-1}, \end{cases} \tag{42}$$

$B(t)$, the minimizer of J_2 with C_i's defined in (29), can be written as (30). Our goal is to find Δ, when $B(t)$ satisfies (42). Similar to the previous subsection, we can show that Δ satisfies

$$\sum_{i=1}^{N} \big(A^TA(\Delta\Lambda_i + \Lambda_i\Delta^T)A^TA\Lambda_i\big)^\circ = -\sum_{i=1}^{N} \big(A^TA\big(A^{-1}N_i(A^{-1})^T\big)^\circ A^TA\Lambda_i\big)^\circ. \tag{43}$$

Presence of the terms A^TA in (43) makes the decoupling that we saw in (31) not possible here. Note that in (31) the effect of A and Λ_i are separated very much, but in (43) this is not the case. This is because J_2 is not congruence preserving, i.e., it is not expressed in terms of only the BC_iB^T's. Note that if A is close to a diagonal multiple of an orthogonal matrix, i.e., if $A \approx QD$, where D is a nonsingular diagonal matrix and Q is orthogonal, then $A^TA \approx D^TD$; and as a result, we have that (43) reduces to

$$\sum_{i=1}^{N} \big(\Delta\Lambda_i + \Lambda_i\Delta^T\big)\Lambda_i \approx -\sum_{i=1}^{N} \big(A^{-1}N_i(A^{-1})^T\big)^\circ\Lambda_i, \tag{44}$$

which is the approximated version of (31). This case is of practical interest. Many algorithms for NOJD try to iteratively reduce the data matrices C_i's, by congruence transforms, to diagonal matrices, i.e., $C_i \leftarrow B_kC_iB_k^T$, where B_k is the local joint diagonalizer found at step k. After a number of iterations, and when the matrices under transformation become close to diagonal, we have $C_i = BA\Lambda_i(BA)^T + tBN_iB^T$, where B is the product of the local joint diagonalizers. In this case, the new A ($A \leftarrow BA$) is close to diagonal, so is A^TA. Also, in another case if one of the C_i's, say C_1, is positive definite, then we can apply the transformation $C_i \leftarrow C_1^{-\frac{1}{2}}C_i(C_1^{-\frac{1}{2}})^T = C_1^{-\frac{1}{2}}A\Lambda_i(C_1^{-\frac{1}{2}}A)^T + tC_1^{-\frac{1}{2}}N_i(C_1^{-\frac{1}{2}})^T$, where $C_1^{\frac{1}{2}}$ is a square root of C_1, i.e., $C_1^{\frac{1}{2}}(C_1^{\frac{1}{2}})^T = C_1$. Again, we can show that if the noise is not too strong, for the new A ($A \leftarrow C_1^{-1/2}A$), we have $A \approx QD$, for some orthogonal Q and nonsingular diagonal D. This case can, for example, correspond to the so-called prewhitening step in the ICA problem, where C_1 is a covariance matrix. We can maintain that the sensitivity properties of NOJD based on J_2 are very similar to those of NOJD based on J_1.

4.3. Sensitivity analysis for NOJD based on J_3. A stationary point $B(t)$ of J_3, when C_i's are of the form (29) and Λ_i's are positive definite, satisfies (28). Similar to previous derivations, by differentiating (28) with respect to t and considering (29)

and (30), we have that Δ, with $\mathrm{diag}(\Delta)=0$, satisfies

$$\sum_{i=1}^{N}\Delta+\Lambda_i\Delta^T\Lambda_i^{-1}=-\sum_{i=1}^{N}\left(A^{-1}N_i(A^{-1})^T\right)^{\circ}\Lambda_i^{-1}. \tag{45}$$

We have used the fact $\frac{d}{dt}X^{-1}=-X^{-1}(\frac{d}{dt}X)X^{-1}$, where X is a nonsingular differentiable matrix function of t. Let us define

$$\tau_{kl}=\frac{1}{N}\sum_{i=1}^{N}\frac{\lambda_{ik}}{\lambda_{il}},\quad \mu_{kl}=\tau_{kl}\tau_{lk}=\frac{1}{N^2}\left(\sum_{i=1}^{N}\frac{\lambda_{ik}}{\lambda_{il}}\right)\left(\sum_{i=1}^{N}\frac{\lambda_{il}}{\lambda_{ik}}\right). \tag{46}$$

Also let:

$$\mathcal{S}=\textstyle\sum_{i=1}^{N}\left(A^{-1}N_i(A^{-1})^T\right)^{\circ}\Lambda_i^{-1},\quad H_{kl}=N\begin{bmatrix}1 & \tau_{kl}\\ \tau_{lk} & 1\end{bmatrix}. \tag{47}$$

Here $\mathcal{S}$ is very similar to $\mathcal{T}$ and represents possible noise amplification due to small norm or ill-conditioning of A. The structure of $\mathcal{S}$ also shows that if Λ_i's are close to singularity, then noise amplification can happen. Note that the cost function J_3 requires Λ_i's to be positive definite, and as this condition is close to violation (by one of Λ_i's being almost singular), then J_3 becomes very sensitive to noise. Equation (45) decouples as

$$H_{kl}\begin{bmatrix}\Delta_{kl}\\ \Delta_{lk}\end{bmatrix}=-\begin{bmatrix}\mathcal{S}_{kl}\\ \mathcal{S}_{lk}\end{bmatrix},\ 1\le k<l\le n, \tag{48}$$

or, equivalently,

$$\begin{bmatrix}\Delta_{kl}\\ \Delta_{lk}\end{bmatrix}=\frac{1}{N(\mu_{kl}-1)}\begin{bmatrix}1 & -\tau_{kl}\\ -\tau_{lk} & 1\end{bmatrix}\begin{bmatrix}\mathcal{S}_{kl}\\ \mathcal{S}_{lk}\end{bmatrix},\ 1\le k<l\le n. \tag{49}$$

It is easy to see that $\|H_{kl}^{-1}\|_F^2=\frac{2+\tau_{kl}^2+\tau_{lk}^2}{(N(\mu_{kl}-1))^2}$, and hence σ_{max}, the larger singular value of H_{kl}^{-1}, satisfies

$$\frac{1}{\sqrt{2}}\frac{\sqrt{\tau_{kl}^2+\tau_{lk}^2+2}}{N(\mu_{kl}-1)}\le\sigma_{max}<\frac{\sqrt{\tau_{kl}^2+\tau_{lk}^2+2}}{N(\mu_{kl}-1)}. \tag{50}$$

From (49) and the previous bound we have

$$\left\|\begin{bmatrix}\Delta_{kl}\\ \Delta_{lk}\end{bmatrix}\right\|\le\sigma_{max}\left\|\begin{bmatrix}\mathcal{S}_{kl}\\ \mathcal{S}_{lk}\end{bmatrix}\right\|<\frac{\sqrt{\tau_{kl}^2+\tau_{lk}^2+2}}{N(\mu_{kl}-1)}\left\|\begin{bmatrix}\mathcal{S}_{kl}\\ \mathcal{S}_{lk}\end{bmatrix}\right\|. \tag{51}$$

It is also easy to establish the following bound:

$$\|\Delta\|_F<\frac{\beta}{N(\mu-1)}\|\mathcal{S}\|_F\le\frac{n\beta\|A^{-1}\|_2^2}{N(\mu-1)}\sum_{i=1}^{N}\|N_i\|_2\|\Lambda_i^{-1}\|_2, \tag{52}$$

where $\beta=\max_{k\neq l}\sqrt{\tau_{kl}^2+\tau_{lk}^2+2}$ and $\mu=\min_{k\neq l}\mu_{kl}$. In summary we have the following theorem.

THEOREM 4. *Let $C_i=A\Lambda_iA^T+tN_i, 1\le i\le N$ ($t\in[-\delta,\delta]$) with Λ_i's positive definite. Let us define $B(t)$, the nonorthogonal joint diagonalizer for $\{C_i\}_{i=1}^N$, as the*

minimizer of J_3. Then for small enough δ the joint diagonalizer can be written as $B(t) = (I + t\Delta)A^{-1} + o(t)$, where Δ (with diag$(\Delta) = 0$) satisfies (48) *as well as* (52).

Note that here, similar to the case of NOJD based on J_1, the modulus of uniqueness (μ) and the condition number of A affect the sensitivity. If one of the Λ_i's is close to singularity, i.e., if $\|\Lambda_i^{-1}\|$ is large, then the NOJD problem can be ill-conditioned. Therefore, almost similar to section 4.1, we might impose the constraint $\|N_i\|_2 \leq \|A\|_2^2 \|\Lambda_i^{-1}\|_2^{-1}$ and define the condition number for the NOJD problem based on J_3 as $\frac{\text{cond}(A)^2}{\mu - 1}$. The imposed condition simply means that if $\|\Lambda_i^{-1}\|_2$ is large, then $\|N_i\|_2$ must be small or $\|A\|_2$ should be large.

4.4. Effect of the number of matrices. One of our motivations in performing sensitivity analysis for the problem of NOJD has been to consider the effect of the number of matrices on the accuracy of the solution. $N = 2$ matrices are enough to find a unique nonorthogonal joint diagonalizer if $|\rho| < 1$. However, to combat noise, we may want to include more matrices. Inclusion of more matrices can have two effects: first on how $\mathcal{T}$ in (41) or $\mathcal{S}$ in (52) changes, second on how ρ, γ and α in (41), or on how μ and α in (52) may change. The first effect is related to noise cancellation through averaging; and the second one is related to improvement of uniqueness measures. Both, of course, depend on how N_i's and Λ_i's are statistically distributed. Let us consider a J_1-based-NOJD problem. Assume that the elements of N_i's are i.i.d. with zero mean, and that the elements of Λ_i's are i.i.d. with mean m and variance σ^2. Also, assume that the matrices are independent from each other. Then, by the strong law of large numbers, we have that $\|\frac{\mathcal{T}}{N}\| \to 0$, $\rho \to \frac{m^2}{\sigma^2 + m^2}$, and $N\alpha \to \frac{2}{\sigma^2 + m^2} < \infty$ as $N \to \infty$ with probability one. Hence, $\|\Delta\|_F \to 0$ as $N \to \infty$ with probability one. Note that this might not happen if N_i's and Λ_i's are of nonzero mean. For small values of N such as $N = 2, 3$, or 4, and, especially when n is large, $|\rho|$ can be very close to unity (for $N = 1$, $|\rho| = 1$). Moreover, the cancellation or averaging effect that we expect to happen in $\mathcal{T}$ for large values of N is not likely to happen for small N. Hence, for small N the NOJD problem can be very sensitive.

4.4.1. More on the number of matrices and modulus of uniqueness. Our claim that for small N the modulus of uniqueness $|\rho|$ can be close to unity deserves more elaboration. From Definition 1, we can interpret ρ_{kl} as the cosine of the angle between two N dimensional vectors $(\lambda_{1k}, \ldots, \lambda_{Nk})^T$ and $(\lambda_{1l}, \ldots, \lambda_{Nl})^T$. Without loss of generality, we can assume that the vectors are of unit length, i.e., they represent points on the unit sphere in $\mathbb{R}^N$. Now $|\rho|$ is the maximum of the absolute value of the cosine of the angles between n points on the unit sphere in $\mathbb{R}^N$. Since $|\rho|$ is independent of the direction of the vectors, we can assume that all the points lie on the same hemisphere on the unit sphere in $\mathbb{R}^N$. The fact that $|\rho|$ can be large when n is much larger than N is related to the fact that among n points on the unit hemisphere in $\mathbb{R}^N$ at least two of them cannot be very far apart from each other. In other words, there are at least two of the points which are closer to each other than a deterministic distance, which depends on n and N. As n increases and the points become denser this deterministic distance decreases and $|\rho|$ approaches unity. Note that $|\rho|$ can become large if there is a large obtuse angle (an angle with negative cosine) between two of the points as well. However, we can only argue that as the number of points increases, they should become denser at some region, and hence, the upper bound on the minimum angular distance between them should decrease. We cannot account for a lower bound on the maximum obtuse angles between the points

unless we know a specific distribution for the points.[7]

Unfortunately, finding the mentioned deterministic bound is difficult for arbitrary N. However, for $N = 2$ it is surprisingly easy to find. Assume that we have $n \geq 3$ points on the unit semicircle. Then we can divide the circumference of the semicircle into $n-1$ arcs each of length $\frac{\pi}{n-1}$. Then by putting n points on the unit semicircle at least two of them will lie on the same arc. Hence, for $N = 2$ we should have $|\rho| \geq \cos\frac{\pi}{n-1}$, for $n \geq 3$. This implies that for two 20×20 matrices, $|\rho| > 0.98$. Of course, for typical matrices this can be worse, since the bound we found is a lower bound. Again, note that this bound is solely based on an upper bound on the minimum of angular distances between the points. In fact, the configuration that achieves the bound $\frac{\pi}{n-1}$ has two antipodal points for which $\rho = -1$. Therefore, the bound is conservative. With more information about the points, we can find better lower bounds. For example, if the matrices are positive definite (i.e., $\lambda_{ik} > 0$ and hence $\rho > 0$), then we can consider points on a quarter of a circle and hence have a lower bound of $\cos\frac{\pi}{2(n-1)}$ on ρ. As a result, for two positive definite matrices of dimension only $n = 10$ we have that $\rho > 0.98$. Maybe the most interesting finding of this paper is that the NOJD of two matrices, if their dimension is fairly large, is ill-conditioned. This is true despite the fact that, as explained before, we might be able to find an exact nonorthogonal joint diagonalizer for the two matrices. Therefore, in general it is better to use more matrices not only to combat noise but also to improve the sensitivity.

The problem of finding an upper bound for the minimum distance between n points on the unit sphere in $\mathbb{R}^N$ is an old problem in the set of problems known as sphere packing problems. Tight bounds for these problems are in general very difficult to find. This specific problem is known as Tammes' problem or "dictators on a planet problem" [12]. One well-known result about it is a bound for $N = 3$. According to this result [24], for $n \geq 3$ points on the unit sphere in $\mathbb{R}^3$, there are at least two points whose spherical (angular) distance is smaller than $d_n = \cos^{-1}\frac{\cot^2\omega_n - 1}{2}$, where $\omega_n = \frac{n}{n-2}\frac{\pi}{6}$. Unfortunately, the proof for this result does not allow an extension to a parallel result for points on the hemisphere. However, we might argue, via homogeneity, that an approximate bound for n points on the hemisphere can be obtained by setting $2n$ points on the sphere and using d_{2n}. Hence, we have $\frac{\cot^2\omega_{2n}-1}{2}$ as an approximate lower bound for $|\rho|$ with $N = 3$. If we ignore the effect of the edge of the hemisphere, this scaling argument sounds quite plausible. Note that the scaling argument becomes more plausible for dense points. We expect the lower bound on $|\rho|$ to be smaller for $N = 3$ than that for $N = 2$, with equal n; and this is exactly what we observe. In Figure 2 we have plotted four curves. The lower two curves show the deterministic lower bounds on $|\rho|$ for $N = 2$ and $N = 3$ in terms of n. As can be seen, the bound for $N = 2$ is higher than the one for $N = 3$. The upper two curves show the average $|\rho|$ in terms of n, this time, for the uniform distribution of points on the circle and sphere. By uniform distribution on the sphere we mean that if $0 \leq \phi < 2\pi$ and $0 \leq \theta \leq \pi$ are the spherical coordinates of a point on the sphere, then these two random variables are uniformly distributed on their domains. Therefore, we generate n points (on the

[7] A more natural framework is to equip the unit sphere with a quotient topology and to identify its antipodal points, which results in the real projective plane in $\mathbb{R}^N$. This way we automatically take into account both the small acute and the large obtuse angles. While this paper was at the final stages of publication, the author realized that a more general version of the problem described previously, rather recently, has been addressed in the literature under the name of "Grassmannian packing problem." Nevertheless, the simplistic analysis presented here seems to be adequate for our main purposes.

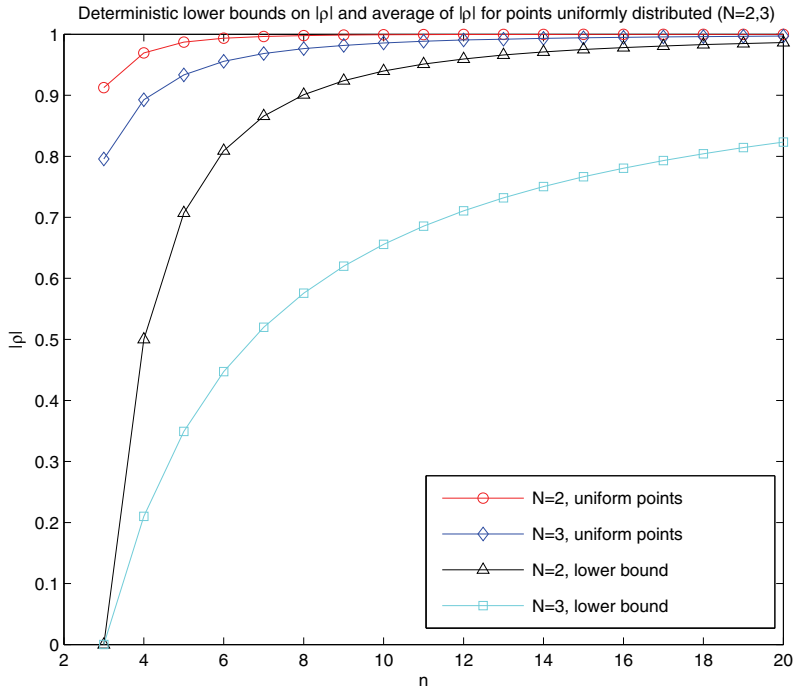


FIG. 2. *This graph shows two forms of variation of* $|\rho|$ *in terms of* n *for* $N = 2$ *and* $N = 3$. *The higher two curves show the average* $|\rho|$ *for points that are uniformly distributed on the circle and sphere. Here uniform means that the angular coordinates of the points in the spherical coordinate are uniformly distributed over the appropriate ranges. The lower two curves are the deterministic lower bounds described in the text. The deterministic bound for* $N = 2$ *is higher than the one for* $N = 3$, *as expected.*

circle or the sphere), find the $|\rho|$ for them, repeat this experiment $10,000$ times, and find the average $|\rho|$. As can be seen, these values of $|\rho|$ are much higher than the bounds.

How about for other values of N? Let us pretend that we could extend the simple argument for the circle to higher dimensions. This helps us unveil the main dynamics between n and N in affecting $|\rho|$. Denote the surface area of the unit sphere in $\mathbb{R}^N$ by S_N. Assume we could divide the surface of the hemisphere into $n-1$ congruent hyper-spherical regular polygons. This, of course, is a very difficult assumption to make. Let the angular diameter of each polygon be θ. If n is large, then we can approximate the area of the polygon by $V_{N-1}(\frac{\theta}{2})^{N-1}$, where V_{N-1} is the volume of the unit hyper-sphere in $\mathbb{R}^{N-1}$. Hence, we have $(n-1) \times V_{N-1}(\frac{\theta}{2})^{N-1} \approx \frac{S_N}{2}$, or $\theta \approx 2(\frac{1}{n-1})^{\frac{1}{N-1}}(\frac{S_N}{2V_{N-1}})^{\frac{1}{N-1}}$. One can see (for example, from the explicit formula for the surface and volume of the hyper-sphere in [27] and the formula related to the Gamma function in [26]) that $(\frac{S_N}{2V_{N-1}})^{\frac{1}{N-1}}$ is of order $O(1)$ for large N; and in fact, it converges to 1. Here $O(.)$ is the big O notation. Now, $\theta \approx 2(\frac{1}{n})^{\frac{1}{N}}$ for large n and

N, which in turn implies

$$|\rho| \geq \cos(\theta) \approx 1 - 2\left(\frac{1}{n}\right)^{\frac{2}{N}}. \tag{53}$$

This is in agreement, at least in its form, with a much more rigorous bound given in [12, Equation (66), p. 28]. To be accurate, the result in [12] states that if n is the maximum number of spherical caps of angular diameter $0 < \theta < 63^\circ$ that can be placed on the surface of the unit sphere in $\mathbb{R}^N$ without overlapping, then for large N

$$\cos\theta \gtrsim 1 - \left(\frac{1}{4}\right)^{0.099}\left(\frac{1}{n}\right)^{\frac{2}{N}} \approx 1 - 0.87\left(\frac{1}{n}\right)^{\frac{2}{N}}. \tag{54}$$

We can replace n with $2n$ to have a similar (approximate) result for the hyper-hemisphere, which essentially does not change the asymptotic bound. Either bounds suggests that in order to control $|\rho|$, as n increases, it suffices to have $N = O(\log n)$, which is encouraging! As a result, we do not need to have too many matrices in order to avoid the ill-conditioning that happens due to a small number of matrices being used. Note that if there is a structural cause of ill-conditioning within Λ_i's, then this recipe is irrelevant. Also, note that in (54) for fixed n as N increases $|\rho|$ does not decrease indefinitely; and there is an asymptotic nonzero lower bound of 0.13. Unfortunately, our approximate bound in (53) does not give an interesting answer in this case. The mentioned behavior is observed in our simulations. Figure 3 shows the experimental and fitted behavior of ρ in terms of N for $n = 20$. The experimental ρ comes from generating λ_{ik}'s independently from uniform distribution on $[0, 1]$. Each value of ρ is an average over 1000 runs. The graph also shows the curve $\tilde{\rho} = 1 - 0.20(\frac{1}{n})^{\frac{5.59}{N}}$ (with $n = 20$), which is fitted to the experimental data. These two curves obviously demonstrate the predicted dynamics between n and N in determining ρ. The mentioned asymptotic lower bound for ρ as $N \to \infty$ in this case is 0.8. The interesting point is that, for small N improvement of ρ is dramatic as N increases, whereas for larger N and better ρ increasing N does not improve the sensitivity significantly. Recall that the important quantity in the sensitivity is $\frac{1}{1-\rho^2}$ which drops rapidly at the first few N's. In fact for the experimental data it drops from 10^4 at $N = 2$ to 8.6 at $N = 10$. In this case the NOJD of only ten 20×20 matrices is well-conditioned or safe. Of course, use of more matrices improves the answer via averaging out the noise.

The preceding discussion concerned the behavior of $|\rho|$ in terms of N and n. Unfortunately, a similar framework and analysis for μ does not seem obvious. Nevertheless, simulations show that, expectedly, whenever $\rho \approx 1$, μ is also close to unity. Therefore, our conclusion that "for small N and large n, the NOJD problem is ill-conditioned" stays valid when J_3 is used, as well.

5. Numerical experiments. In this section we perform some experiments to examine the derived results. The first example is just a toy example and the second one is more realistic in the context of blind source separation (BSS).

5.1. Example 1. We investigate the effect of ρ, N, and the condition number of A, $\mathrm{cond}(A)$, on the sensitivity of NOJD for matrices generated as in (29). We generate $\{\Lambda_i\}_{i=1}^N$ with elements that are i.i.d. exponential random variables with mean 1. We choose $n = 10$. We also generate $A_{n\times n}$ randomly. Note that with probability one the joint diagonalizer for $\{C_i\}_{i=1}^N$ is unique. The noise matrices are with standard normal elements.

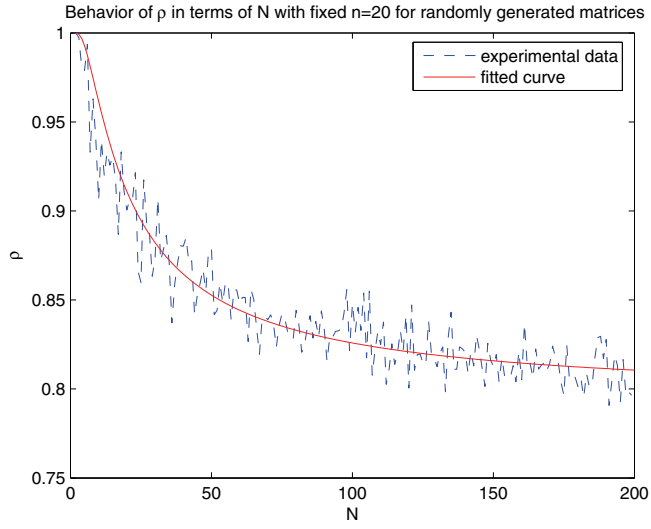


FIG. 3. *A typical behavior of* $|\rho|$ *in terms of* N *for fixed* n. *Here* $n = 20$ *and the* λ_{ik}*'s are generated from a uniform distribution on* $[0, 1]$. *The dashed curve shows the experimental* ρ. *Each point is an average over* 1000 *runs. The solid curve shows the curve* $\tilde{\rho} = 1 - 0.20(\frac{1}{n})^{\frac{5.59}{N}}$ *(with* $n = 20$*), which is fitted to the data.*

We consider the quality of joint diagonalization in terms of noise level t, ρ, and the condition number of A. We only consider J_1 and J_2 based methods. We use the QRJ2D algorithm,[8] introduced in [3], to find B and measure the error by

$$\text{Index}(P) = \sum_{i=1}^{n}\left(\sum_{j=1}^{n}\frac{|p_{ij}|}{\max_k |p_{ik}|} - 1\right) + \sum_{j=1}^{n}\left(\sum_{i=1}^{n}\frac{|p_{ij}|}{\max_k |p_{kj}|} - 1\right) \tag{55}$$

with $P = BA$. $\text{Index}(BA) \geq 0$ and the equality happens only when $BA = \Pi D$ (and hence $B = \Pi D A^{-1}$) for some Π and D. The smaller the index is, the better joint diagonalization is, in the sense that B is closer to A^{-1}. We try different values of N (and hence ρ). We also investigate the effect of cond(A), by keeping the Λ_i's the same, and increasing cond(A) while $\|A\|_F$ is constant. Table 5.1 gives the results. The left subtable shows the index for different values of N (hence ρ) for two different noise levels $t = 0$ and $t = 0.01$. By increasing the number of matrices and hence improving the modulus of uniqueness, the sensitivity improves. Note that for $N = 2$, sensitivity is so high that the QRJ2D algorithm does not give a good answer even at zero noise. The right subtable also shows the sensitivity degradation that happens because of increasing cond(A). In this experiment $\|A\|_F = 1$, $N = 100$, $t = 0$, and 0.0001. Sensitivity increases as conditioning of A degrades. Although the actual error values depend on the specific algorithm used, the trend of the error values as the parameters change gives an insight as to what factors affect the sensitivity.

5.2. Example 2: Separation of nonstationary sources. Now we consider a more realistic situation, which is the separation of nonstationary sources using NOJD

[8]The MATLAB code for this algorithm is available online at http://www.isr.umd.edu/Labs/ISL/ICA2006/.

TABLE 5.1

Left: Sensitivity of Index(BA) with respect to noise level t as N, and, hence ρ changes in Example 1. cond(A) = 25.11. Right: Sensitivity of Index(BA) with respect to noise level t as cond(A) increases and $\|A\|_F = 1$ ($\rho = .68$).

Index(BA)	$t = 0$	$t = 0.01$
$N = 2, \rho = 0.9999$	3.9	17.0
$N = 4, \rho = 0.9959$	0.00	3.46
$N = 10, \rho = 0.9662$	0.00	1.46
$N = 100, \rho = 0.6903$	0.00	0.29
$N = 200, \rho = 0.60$	0.00	0.19

Index(BA) ($N = 100, \rho = .68$)	t=0	t=0.0001
cond(A) = 1	0.00	0.01
cond(A) = 2	0.00	0.01
cond(A) = 10	0.00	0.12
cond(A) = 50	0.00	3.02
cond(A) = 100	0.00	28.51

of correlation matrices. This example also allows us to compare NOJD based on J_1 and J_3. The idea of using nonstationarity to separate sources has been described in [21]. Consider model (1) where the source vector is a Gaussian vector of independent components. Also, assume that the sources are nonstationary with varying variances. $R_{\mathbf{xx}}(t_i)$ the correlation matrix of the mixture at time t_i is

$$R_{\mathbf{xx}}(t_i) = A\Lambda_s(t_i)A^T, \tag{56}$$

where $\Lambda_s(t_i)$ is the (diagonal) correlation matrix of the source at time t_i. Suppose that we gather the correlation matrices at times $t_1, ..., t_N$, and form the set $\{R(t_i)\}_{i=1}^N$. If $\Lambda_s(t_i)$ changes enough such that the modulus of uniqueness for this set is smaller than one, then NOJD of this set yields an estimation for A^{-1}; and hence, it can result in the separation of the mixture.

We have $n = 10$ sources. First we generate a random matrix $A_{n\times n}$. The condition number for this matrix is 75.11. Then we generate the sources as follows. We assume that the sources are stationary on short periods of $T = 100$ samples, and that they change their variances randomly at the end of each period. We consider $N = 20$ periods. During the ith stationary period, the jth source has Gaussian distribution with zero mean and a random variance λ_{ij}. We draw each random standard deviation $\sqrt{\lambda_{ij}}$ from a uniform distribution on $[0, 1]$. Also, during the stationary periods each source generates independent samples. The sources are mixed through A. In each stationary period, we use the observed 100 samples of the mixture to estimate the correlation matrix for the mixture in that period. After the first stationary period, at the end of each stationary period, we perform an NOJD of the estimated correlation matrices gathered up to that time, in order to estimate A^{-1}. Also, as time passes, we compute ρ and μ for the set of true correlation matrices based on λ_{ij}'s. We use three different methods for NOJD of the estimated correlation matrices: (i) Pham's algorithm [20] which uses J_3 and requires positive definite matrices, (ii) QRJ2D algorithm, which is based on J_2, and (iii) FFDIAG [29] which is based on J_1. As we mentioned before, J_1 and J_2 based NOJD have similar sensitivity properties. We use QRJ2D and FFDIAG, since we want to have more evidence for comparing J_1-based-NOJD and J_3-based-NOJD. The output of each of these algorithms is an unmixing matrix B. In order to measure the performance we use two measures. One is Index(BA), which we introduced before. The other one is the mean-squared interference to signal ratio (ISR), which measures how much other sources are present at each restored source. Note that from (30) for the recovered source vector $\vec{\mathbf{y}}$ we have

$$\vec{\mathbf{y}} = B(t) \qquad \vec{\mathbf{x}} = B(t) \qquad \vec{\mathbf{s}} \approx \vec{\mathbf{s}} + t\Delta\vec{\mathbf{s}}. \tag{57}$$

Here again, we have ignored the possible scaling and permutation ambiguity in the

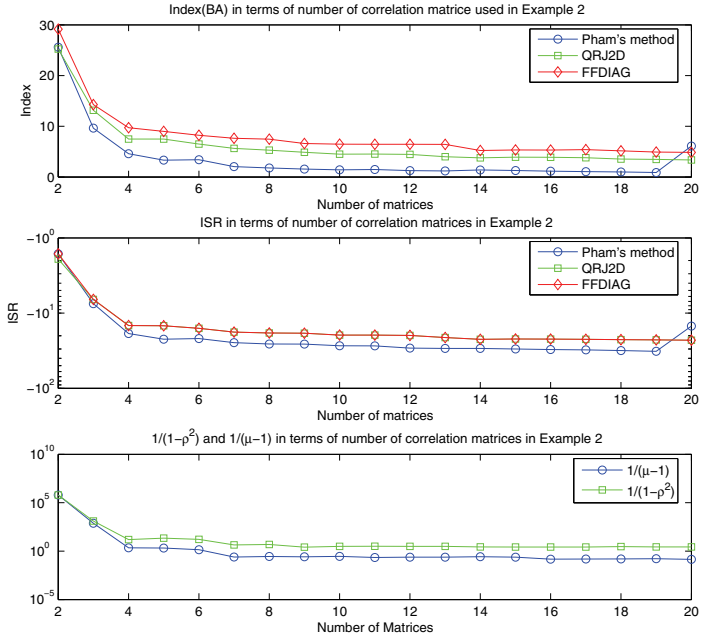


FIG. 4. *This figure shows the performance of source separation for nonstationary sources based on NOJD of correlation matrices at different times. As time passes, more correlation matrices are used. Three different methods for NOJD are employed: Pham'a algorithm, which is based on J_3, QRJ2D algorithm, which uses J_2, and FFDIAG, which uses J_1. Top: Index(BA) in terms of the number of correlation matrices used. Middle: ISR in terms of the number of correlation matrices used. Bottom: $\frac{1}{1-\rho^2}$ and $\frac{1}{\mu-1}$ in terms of the number of correlation matrices used. The jump seen at $i = 20$ in the graphs for J_3-based-NOJD is because in the last period some of the sources become extremely weak and the correlation matrix for that period becomes almost singular.*

restored vector. In practice, of course, we compute Δ from $P = BA$, after reordering and normalizing the rows of P. As the above equation suggests, $\|\Delta\|_F$ also measures the mean-squared ISR,[9] i.e., how much interference from other sources is present in each recovered source. We use

$$\text{ISR} = 10 \log \frac{\|\Delta\|_F^2}{n} \tag{58}$$

as a measure of the interference. Note that in this example we have no noise and the source of error is the estimation error due to a finite number of data samples. Another point that we want to examine is the sensitivity of NOJD based on J_3, in the case when one of the matrices becomes almost singular. For that purpose, in the last ($i = 20$) interval we set the standard deviation of six of the sources to 10^{-10}.

Figure 4 shows the results of the experiment. The top graph shows the Index(BA) in terms of i (which is the number of correlation matrices used) for different methods. The middle graph gives the ISR in terms of i. Note that the ISR measures for QRJ2D and FFDIAG are very close; however, the index measure for these two methods differ. The bottom graph shows $\frac{1}{1-\rho^2}$ and $\frac{1}{\mu-1}$ in terms of i for the correlation matrices

[9]The author is thankful to one of the anonymous reviewers for a reminder of this observation.

involved. As explained before, these two numbers, in fact, can be considered as condition numbers for NOJD based on J_1 and J_3, respectively (we have omitted the effect of A, i.e., $\mathrm{cond}(A)^2$, which is common in both condition numbers; see the last paragraphs of sections 4.1 and 4.3). For $i = 2, 3$ the numbers are very high for both of the cases. One can see that after the first few i, the condition numbers do not improve much. Note that the condition number for the J_1-based-NOJD is higher than that of the J_3-based-NOJD; and at the same time the J_3-based-NOJD yields better separation except for $i = 2$ and $i = 20$. From our theoretical results this is certainly what we expect. However, we cannot relate these two facts immediately, since the actual numbers depend on many factors. Note that NOJD for $i = 2, 3$ is not so effective, since the ISR measure is really poor (around or above -7dB). As a comparison, the best ISR that Pham's method achieves is -32dB, and the best one that QRJ2D or FFDIAG achieve is about -23dB. The jumps at $i = 20$ in $\mathrm{Index}(BA)$ and in ISR for Pham's method are due to the fact that the last correlation matrix is almost nonsingular. As we can see, the NOJD based on J_1 gives better separation in this case. Recall that J_1 does not require nonsingular matrices, and the condition number we assigned for J_3-based-NOJD was based on the assumption that $\|\Lambda_i^{-1}\|_2$'s are not too large. At $i = 20$ this condition is violated and that is why despite the fact that $\frac{1}{\mu-1} < \frac{1}{1-\rho^2}$, the J_1-based-NOJD performs better. For the curious reader, we mention that despite some evidence in this example, for a given $\{\Lambda_i\}_{i=1}^N$ the conjecture that $\frac{1}{\mu_{kl}-1} \leq \frac{1}{1-\rho_{kl}^2}$ or $\frac{1}{\mu-1} \leq \frac{1}{1-\rho^2}$ is not true. However, note that for $\frac{1}{\mu-1} \leq \frac{1}{1-\rho^2}$ to hold it is sufficient to have $\mu \geq 2$, which can be achieved since the range for μ is the long half-line $[1, +\infty)$. This may explain why in most simulations and in this example $\frac{1}{\mu-1} < \frac{1}{1-\rho^2}$.

6. Conclusions. We introduced the NOJD problem and the related EJD problem. We derived the uniqueness conditions for the EJD problem. We gave a joint diagonalization based formulation of ICA. Factors that affect the sensitivity of the NOJD problem were investigated. Modulus of uniqueness captures the uniqueness of the exact joint diagonalization problem and it affects the sensitivity of the NOJD problem that arises from adding noise to clean matrices. Also we showed that if the sought joint diagonalizer is ill-conditioned, then sensitivity will be high. We tried to quantitatively show how dimension of the matrices and the number of matrices can affect the modulus of uniqueness. In particular, we showed that the NOJD problem can be very ill-conditioned if the number of matrices is small and they are fairly large. Sensitivity of the NOJD problem depends on the cost function used; and in one example we gave a comparison of the behaviors of two different cost functions for NOJD.

Acknowledgments. The framework outlined in section 3.2.1 to derive the non-holonomic flow for NOJD has been conveyed to the author by Prof. P. S. Krishnaprasad. The author also would like to thank him for his support of this work. The author feels indebted to the anonymous reviewers who have read this paper and whose insightful suggestions have helped in improving the work. Also, the author cordially thanks Dr. Mehdi Kalantari for his comments and suggestions to improve this paper.

REFERENCES

[1] P.-A. Absil and K. A. Gallivan, *Joint diagonalization on the oblique manifold for independent component analysis*, in Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006), Toulouse, France, 2006.

[2] B. Afsari, *Gradient Flow Based Matrix Joint Diagonalization for Independent Component Analysis*, Master's thesis, Univeristy of Maryland, College Park, MD, 2004.
[3] B. Afsari, *Simple LU and QR based non-orthogonal matrix joint diagonalization*, in ICA, J. P. Rosca, D. Erdogmus, J. C. P., and S. Haykin, eds., Lecture Notes in Comput. Sci. 3889, Springer, 2006, pp. 1–7.
[4] B. Afsari, *What can make joint diagonalization difficult?*, in Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP07), Vol. 3, Honolulu, HI, 2007, IEEE, pp. III–1377–III–1380.
[5] B. Afsari and P. S. Krishnaprasad, *A Novel Nonorthogonal Joint Diagonalization Cost Function for ICA*, Tech. report, Institute for Systems Research, University of Maryland, College Park, MD, 2005.
[6] B. Afsari and P. S. Krishnaprasad, *Some gradient based joint diagonalization methods for ICA*, in ICA, C. G. Puntonet and A. Prieto, eds., Lecture Notes in Comput. Sci. 3195, Springer, 2004, pp. 437–444.
[7] A. Belouchrani, K. A. Merai m, J.-F. Cardoso, and E. Moulines, *A blind source separation technique using second-order statistics*, IEEE Trans. Signal Processing, 45 (1997), pp. 434–444.
[8] J. F. Cardoso, *Perturbation of Joint Diagonalizers*, Tech. report 94D023, Telecom Paris, Paris, France, 1994.
[9] J. F. Cardoso and A. Soloumiac, *Blind beamforming for non-Gaussian signals*, IEEE Proceedings-F, 140 (1993), pp. 362–370.
[10] P. Comon, *Independent component analysis: A new concept?*, Signal Process., 36 (1994), pp. 287–314.
[11] P. Comon, *Canonical Tensor Decompositions*, Tech. report, Lab. I3S, Sophia-Antipolis, France, 2004.
[12] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices, and Groups*, Springer, 1999.
[13] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*, Ph.D. thesis, Katholiecke University, Leuven, Belgium, 1997.
[14] L. De Lathauwer, *A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666.
[15] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1996.
[16] R. Harshman, *Foundations of the PARAFAC procedure: Model and conditions for an explanatory multi-mode factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.
[17] U. Helmke and J. B. More, *Optimization and Dynamical Systems*, Springer-Verlag, London, 1994.
[18] A. M. Kagan, Y. V. Linnik, and C. R. Rao, *Characterization Problems in Mathematical Statistics*, Wiley, New York, 1973.
[19] E. Moreau, *A generalization of joint-diagonalization criteria for source separation*, IEEE Trans. Signal Process., 49 (2001), pp. 530–541.
[20] D. T. Pham, *Joint approximate diagonalization of positive definite Hermitian matrices*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1136–1152.
[21] D.-T. Pham and J. F. Cardoso, *Blind separation of instantaneous mixtures of nonstationary sources*, IEEE Trans. Signal Process., 49 (2001), pp. 1837–1848.
[22] G. W. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
[23] J. M. F. ten Berge and N. D. Sidiropoulos, *On uniqueness in CANDECOMP/PARAFAC*, Psychometrika, 67 (2002), pp. 399–409.
[24] L. Fejes Tóth, *On the densest packing of spherical caps*, American Mathematical Monthly, 56 (1949), pp. 330–331.
[25] A. Van Der Veen, *Joint diagonalization via subspace fitting techniques*, in Proceedings of the 2001 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001), Vol. V, Salt Lake City, UT, 2001, pp. 2773–2776.
[26] E. W Weisstein, *Gamma Function*, From MathWorld, A Wolfram Web Resource, available online at http://mathworld.wolfram.com/GammaFunction.html.
[27] E. W Weisstein, *Hypersphere*, From MathWorld, A Wolfram Web Resource, available online at http://mathworld.wolfram.com/Hypersphere.html.
[28] A. Yeredor, *Nonorthogonal joint diagonalization in the least-squares sense with application in blind source separation*, IEEE Trans. Signal Process., 50 (2002), pp. 1545–1553.
[29] A. Ziehe, P. Laskov, G. Nolte, and K. R. Mueller, *A fast algorithm for joint diagonalization with nonorthogonal transformation and its application to blind source separation*, J. Mach. Learn. Res., 5 (2004), pp. 777–800.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1172–1204

LOWER-RANK TENSOR APPROXIMATION AND MULTIWAY FILTERING*

DAMIEN MUTI†, SALAH BOURENNANE†, AND JULIEN MAROT†

Abstract. This paper presents some recent filtering methods based on the lower-rank tensor approximation approach for denoising tensor signals. In this approach, multicomponent data are represented by tensors, that is, multiway arrays, and the presented tensor filtering methods rely on multilinear algebra. First, the classical channel-by-channel SVD-based filtering method is overviewed. Then, an extension of the classical matrix filtering method is presented. It is based on the lower rank-$(K_1, \ldots, K_N)$ truncation of the higher order SVD which performs a multimode principal component analysis (PCA) and is implicitly developed for an additive white Gaussian noise. Two tensor filtering methods recently developed by the authors are also overviewed. The first method consists of an improvement of the multimode PCA-based tensor filtering in the case of an additive correlated Gaussian noise. This improvement is specially done thanks to the fourth order cumulant slice matrix. The second method consists of an extension of Wiener filtering for data tensors. The performances and comparative results between all these tensor filtering methods are presented for the cases of noise reduction in color images, multispectral images, and multicomponent seismic data.

Key words. multilinear algebra, tensor decomposition, multiway arrays, lower-rank approximation, filtering

AMS subject classifications. 15A69, 15A18, 49M27

DOI. 10.1137/060653263

1. Introduction.

Tensor data modeling and tensor analysis have been improved and used in several application fields, such as quantum physics, economy, chemometrics, psychology, and data analysis. Nevertheless, only recent studies focus their interest on tensor methods in signal processing applications. Tensor formulation in signal processing has received great attention since the recent development of multicomponent sensors, especially in imagery (color or multispectral images, video, etc.) and seismic fields (antenna of sensors recording waves with polarization properties). Indeed, the digital data obtained from these sensors are fundamentally higher order tensor objects, that is, multiway arrays whose elements are accessed via more than two indexes. Each index is associated with a dimension of the tensor generally called "nth-mode" [13, 14, 28, 29].

In recent decades, the classical algebraic processing methods have been specifically developed for vector and matrix representations. They are usually based on the covariance matrix, the cross-spectral matrix, or, more recently, the higher order statistics. Their overall aim is classically to determine a subspace associated with the signal or the parameters to estimate. They mainly rely on three algebraic tools.

(1) The singular value decomposition (SVD) [18], which is used in principal component analysis (PCA);

(2) Penrose–Moore matrix inversion [18]; and

(3) The matrix lower rank approximation, which, according to the Eckart–Young theorem [15], can be achieved thanks to a simple SVD truncation.

These methods have proved to be very efficient in several applications.

*Received by the editors March 1, 2006; accepted for publication (in revised form) by L. De Lathauwer September 10, 2007; published electronically September 25, 2008.
http://www.siam.org/journals/simax/30-3/65326.html

†Institut Fresnel / CNRS UMR 6133 - Domaine Universitaire de Saint Jérôme, F-13397 Marseille Cedex 20, France (damien.muti@fresnel.fr, salah.bourennane@fresnel.fr, julien.marot@fresnel.fr).

When dealing with multicomponent data represented as tensors, the classical processing techniques consist in rearranging or splitting the data set into matrices or vectors in order for the previously quoted classical algebraic processing methods to be applicable. The original data structure is then built anew, after processing.

In order to keep the data tensor as a whole entity, new signal processing methods have been proposed [35, 36, 37]. Hence, instead of adapting the data tensor to the classical matrix-based algebraic techniques (by rearrangement or splitting), these new methods propose to adapt their processing to the tensor structure of the multicomponent data. This new approach implicitly implies the use of multilinear algebra and mathematical tools that extend the SVD to tensors.

Two main tensor decomposition methods that generalize the matrix SVD have been initially developed to achieve a multimode PCA and recently used in tensor signal processing. They rely on two models, the TUCKER3 model and the PARAFAC model.

The TUCKER3 model [29, 48] was adopted in higher order SVD (HOSVD) [2, 13] and in lower rank-$(K_1, \ldots, K_N)$ tensor approximation [11, 14, 47]. We denote by HOSVD-$(K_1, \ldots, K_N)$ the truncation of HOSVD, performed with ranks $(K_1, \ldots, K_N)$, in modes $1, \ldots, N$, respectively. This model recently has been used as multimode PCA, in seismics for wave separation based on a subspace method, in image processing for face recognition and expression analysis [49, 52], and in noise filtering of color images [36].

The PARAFAC model and the CANDECOMP model were developed in [20] and [10], respectively. In [30] the link was set between CANDECOMP and PARAFAC models. The CANDECOMP/PARAFAC model, referred to as the CP model [25], has recently been applied to the food industry [9], array processing [45], and telecommunications [46].

These two decomposition methods differ in the tensor rank definition on which they are based. The HOSVD-$(K_1, \ldots, K_N)$ and the rank-$(K_1, \ldots, K_N)$ approximation rely on the nth-mode rank definition, that is, the rank of the tensor nth-mode flattening matrix [13, 14]. The rank-$(K_1, \ldots, K_N)$ approximation [14] relies on an optimization algorithm which is initialized by the HOSVD-$(K_1, \ldots, K_N)$ [13]. The rank-$(K_1, \ldots, K_N)$ approximation improves the approximation obtained with the HOSVD-$(K_1, \ldots, K_N)$. It relies on the determination of the signal subspace in every nth-mode of the data tensor and copes with additive white Gaussian noise. The rank-$(K_1, \ldots, K_N)$ approximation provides the best approximation in the sense of least Frobenius norm of the difference between estimated and expected tensors. Nevertheless it assumes a noncorrelated Gaussian noise. To face the case of correlated Gaussian noise, a variant of rank-$(K_1, \ldots, K_N)$ approximation, based on fourth order cumulants, was proposed [39]. Indeed, as it is proved in [33], the fourth order cumulants of a Gaussian variable are null.

A tensor framework was employed by [12] to express the solution to the linear independent component analysis (ICA) problem which employs fourth order cumulants. The multilinear ICA (N-mode ICA) model [50, 51], which was developed for face recognition, encodes the fourth order cumulants for each of the nth-mode flattening matrices of the tensor.

The CP model relies on a canonical decomposition of a tensor into a summation of rank-one tensors and on the extension of the classical matrix rank. Details on the tensor ranks and orthogonal tensor decomposition can be found in [22, 27].

When the TUCKER3 model and the PARAFAC model are associated with an ALS loop, they are known respectively as the TUCKALS3 algorithm [29, 28] and

the PARAFAC ALS algorithm [30, 20]. Many recent studies have been conducted to improve the convergence of these algorithms [14, 26, 56, 44].

The goal of this paper is to present an overview of the principal results concerning this new approach of data tensor filtering. More details on the algorithms presented in this survey can be found in [35, 36, 38, 39]. These algorithms are analogous to multilinear ICA but were developed independently for image filtering. The presented algorithms are based on a signal subspace approach, so they are efficient when the noise components are uncorrelated, when the signal and the additive noise are uncorrelated, and when some rows or columns of the image are redundant. In this case it is possible to distinguish between a signal subspace and a noise subspace, as for the traditional SVD-based filtering and Wiener filtering algorithms. Wiener filtering requires prior knowledge on the expected noisefree signal or image. However, multiway filtering methods provide the following advantage over traditional filtering methods: by apprehending a multiway data set as a whole entity, they take into account the dependence between modes thanks to ALS algorithms. The goal of the paper is also to present some simulations and comparative results concerning color images and multicomponent seismic signal filtering.

The paper is organized as follows. Section 2 presents the tensor data and a short overview of its main properties. Section 3 introduces the tensor formulation of the classical noise-removal problem as well as some new tensor filtering notations. First, we explain how the channel-by-channel SVD-based method processes successively each component of the data tensor. Second, we consider two methods that take into account the relationships between each component of the considered tensor. These two methods are based on the nth-mode signal subspace. The first method for signal tensor estimation is based on multimode PCA achieved by rank-$(K_1, \ldots, K_N)$ approximation. The second method is a new tensor version of Wiener filtering. Section 4 presents some comparative results where the overviewed multiway filtering methods are applied to noise reduction in color images, denoising of multispectral images, and denoising of multicomponent seismic waves. Section 5 concludes the paper.

The following notation is used in the rest of the paper. Scalars are denoted by italic lowercase roman (a); vectors by boldface lowercase roman ($\mathbf{a}$); matrices by boldface uppercase roman ($\mathbf{A}$); and tensors by uppercase calligraphic ($\mathcal{A}$). We distinguish a random vector, like $\mathbf{a}$, from one of its realizations by using a supplementary index, like $\mathbf{a}_i$.

2. Tensor representation and properties. We define a tensor of order N as a multidimensional array whose entries are accessed via N indexes. A tensor is denoted by $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, where each element is denoted by $a_{i_1 \cdots i_N}$, and $\mathbb{R}$ is the real manifold. Each dimension of a tensor is called nth-mode, where n refers to the n^{th} index. Figure 2.1 shows how a color image can be represented by a third order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, where I_1 is the number of rows, I_2 is the number of columns, and I_3 is the number of color channels. In the case of a color image, we have $I_3 = 3$. Let us define $E^{(n)}$ as the nth-mode vector space of dimension I_n, associated with the nth-mode of tensor $\mathcal{A}$. By definition, $E^{(n)}$ is generated by the column vectors of the nth-mode flattening matrix. The nth-mode flattening matrix $\mathbf{A}_n$ of tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is defined as a matrix from $\mathbb{R}^{I_n \times M_n}$, where

$$M_n = I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-1}. \tag{2.1}$$

For example, when we consider a third order tensor, the definition of the matrix flattening involves the dimensions $I_1,\ I_2,\ I_3$ in a backward cyclic way [5, 13, 25].

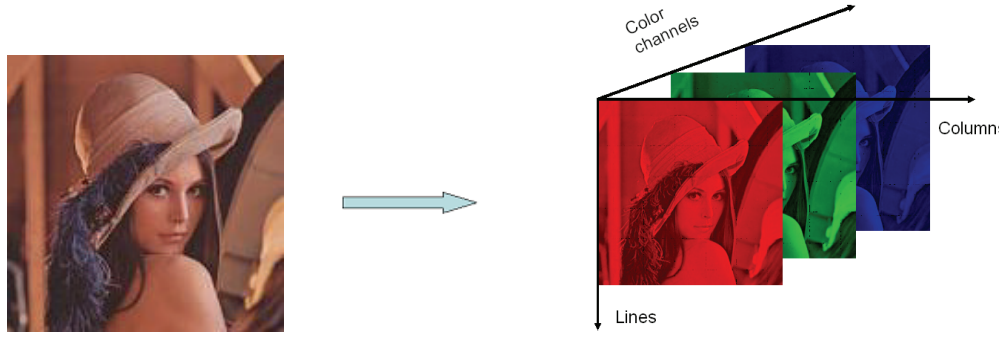


FIG. 2.1. *Lena standard color image and its tensor representation.*

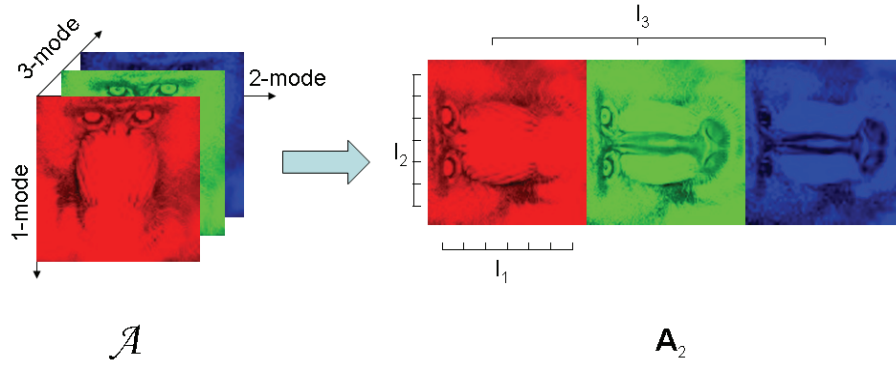


FIG. 2.2. *2nd-mode flattening of tensor* $\mathcal{A}$*:* $\mathbf{A}_2$.

When dealing with a 1st-mode flattening of dimensionality $I_1 \times (I_2 I_3)$, we formally assume that the index i_2 varies more slowly than i_3. For all $n = 1$ to 3, $\mathbf{A}_n$ columns are the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying the index i_n from 1 to I_n and keeping the other indexes fixed. These vectors are called the nth-mode vectors of tensor $\mathcal{A}$. An illustration of the 2nd-mode flattening of a color image is presented in Figure 2.2.

In the following, we use the operator $\times_n$ as the nth-mode product, which generalizes the matrix product to tensors. Given $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$, the nth-mode product between tensor $\mathcal{A}$ and matrix $\mathbf{U}$ leads to the tensor $\mathcal{B} = \mathcal{A} \times_n \mathbf{U}$, which is a tensor of $\mathbb{R}^{I_1 \times \cdots I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$, whose entries are given by

$$b_{i_1 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 \cdots i_{n-1} i_n i_{n+1} \cdots i_N} u_{j_n i_n}. \tag{2.2}$$

The next section presents the recent filtering methods for tensor data.

3. Tensor filtering problem formulation. The tensor data extend the classical vector data. The measurement of a multidimensional and multiway signal $\mathcal{X}$ by multicomponent sensors with additive noise $\mathcal{N}$ results in a data tensor $\mathcal{R}$ such that

$$\mathcal{R} = \mathcal{X} + \mathcal{N}. \tag{3.1}$$

$\mathcal{R}$, $\mathcal{X}$, and $\mathcal{N}$ are tensors of order N from $\mathbb{R}^{I_1 \times \cdots \times I_N}$. Tensors $\mathcal{N}$ and $\mathcal{X}$ represent noise and signal parts of the data, respectively. The goal of this study is to estimate the expected signal $\mathcal{X}$ thanks to a multidimensional filtering of the data [35, 36, 38, 39]:

$$\widehat{\mathcal{X}} = \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \mathbf{H}^{(2)} \times_3 \cdots \times_N \mathbf{H}^{(N)}, \tag{3.2}$$

From a signal processing point of view, the nth-mode product is a nth-mode filtering of data tensor $\mathcal{R}$ by nth-mode filter $\mathbf{H}^{(n)}$. Consequently, for all $n = 1$ to N, $\mathbf{H}^{(n)}$ is the nth-mode filter applied to the nth-mode of the data tensor $\mathcal{R}$.

In this paper we assume that the noise $\mathcal{N}$ is independent from the signal $\mathcal{X}$ and that the nth-mode rank K_n is smaller than the nth-mode dimension I_n ($K_n < I_n$, for all $n = 1$ to N). Then it is possible to extend the classical subspace approach to tensors by assuming that, whatever the nth-mode, the vector space $E^{(n)}$ is the direct sum of two orthogonal subspaces, namely, $E_1^{(n)}$ and $E_2^{(n)}$, which are defined as follows:

- $E_1^{(n)}$ is the subspace of dimension K_n, spanned by the K_n singular vectors associated with the K_n largest singular values of matrix $\mathbf{X}_n$; $E_1^{(n)}$ is called the signal subspace [1, 33, 55, 54].
- $E_2^{(n)}$ is the subspace of dimension $I_n - K_n$, spanned by the $I_n - K_n$ singular vectors associated with the $I_n - K_n$ smallest singular values of matrix $\mathbf{X}_n$; $E_2^{(n)}$ is called the noise subspace [1, 33, 55, 54].

The dimensions $K_1, K_2, \ldots, K_N$ can be estimated by means of the well-known Akaike information criterion (AIC) or Minimum description length (MDL) criteria [53], which are entropy-based information criteria. Hence, one way to estimate signal tensor $\mathcal{X}$ from noisy data tensor $\mathcal{R}$ is to estimate $E_1^{(n)}$ in every nth-mode of $\mathcal{R}$. The following section presents three tensor filtering methods based on nth-mode signal subspaces. The first method is an extension of classical matrix filtering algorithms. It consists of a channel-by-channel SVD-based filtering.

The second filtering method is based on multimode PCA achieved by rank-$(K_1, \ldots, K_N)$ approximation. Two algorithms are presented for this case. The first algorithm is implicitly developed for an additive *white* and Gaussian noise assumption, whereas the second algorithm represents an improvement of the first one in the case of a *correlated* Gaussian noise. This improvement is achieved thanks to higher order statistics.

The third method, the multiway Wiener filtering (Wmm-$(K_1, \ldots, K_N)$), is an algorithm that extends the classical two-dimensional Wiener filtering to tensor data.

3.1. Channel-by-channel SVD-based filtering. The classical algebraical methods operate on two-dimensional data matrices and are based on the SVD [1, 3, 4] and on the Eckart–Young theorem concerning the best lower rank approximation of a matrix [15] in the least-squares sense.

In the first method, a preprocessing is applied to the multidimensional and multiway data. It consists in splitting data tensor $\mathcal{R}$, representing the noisy multicomponent image into two-dimensional "slice matrices" of data, each representing a specific channel. According to the classical signal subspace methods [8], the left and right signal subspaces, corresponding to, respectively, the column and the row vectors of each slice matrix, are simultaneously determined by processing the SVD of the matrix associated with the data of the slice matrix. Let us consider the slice matrix $\mathcal{R}(:, :, i_3, \ldots, i_j, \ldots, i_N)$ of data tensor $\mathcal{R}$. Projectors $\mathbf{P}$ on the left signal subspace and $\mathbf{Q}$ on the right signal subspace are built from, respectively, the left and the right singular vectors associated with the K largest singular values of $\mathcal{R}(:, :, i_3, \ldots, i_j, \ldots, i_N)$.

The parameter K simultaneously defines the dimensions of the left and right signal subspaces. Applying the projectors $\mathbf{P}$ and $\mathbf{Q}$ on the slice $\mathcal{R}(:,:,i_3,\ldots,i_j,\ldots,i_N)$ amounts to computing its best lower rank-K matrix approximation [15] in the least-squares sense.

The filtering of each slice matrix of data tensor $\mathcal{R}$ separately is called in the following "channel-by-channel" SVD-based filtering of $\mathcal{R}$. It consists of a first way to estimate the signal tensor $\mathcal{X}$ and can be summarized by the following steps:

1. input: data tensor $\mathcal{R}$, left and right signal subspace dimension K.
   for $i_N = 1$ to I_N:
   for $i_{N-1} = 1$ to I_{N-1}:
   $\vdots$
   for $i_4 = 1$ to I_4:
   for $i_3 = 1$ to I_3:

   (a) calculate matrix $\mathcal{R}(:,:,i_3,\ldots,i_j,\ldots,i_N)$ SVD:

   $$\mathcal{R}(:,:,i_3,\ldots,i_j,\ldots,i_N) = \mathbf{U}\cdot\mathbf{\Sigma}\cdot\mathbf{V}^T,$$

   where $\mathbf{\Sigma}$ is the core matrix regrouping the singular values of the matrix $\mathcal{R}(:,:,i_3,\ldots,i_j,\ldots,i_N)$, and $\mathbf{U} = [\mathbf{u}_1\ldots\mathbf{u}_{I_1}]$ and $\mathbf{V} = [\mathbf{v}_1\ldots\mathbf{v}_{I_2}]$ are the matrices containing the left and right singular vectors defined respectively by $\mathbf{u}_{i_1}$ and $\mathbf{v}_{i_2}$.
   (b) construct matrices $\mathbf{U}_K = [\mathbf{u}_1\ldots\mathbf{u}_K]$ and $\mathbf{V}_K = [\mathbf{v}_1\ldots\mathbf{v}_K]$ containing the K largest left and right eigenvectors of $\mathcal{R}(:,:,i_3,\ldots,i_j,\ldots,i_N)$;
   (c) compute the projector $\mathbf{P} = \mathbf{U}_K\mathbf{U}_K^T$ on the column signal subspace, and projector $\mathbf{Q} = \mathbf{V}_K\mathbf{V}_K^T$ on the row signal subspace.
   (d) compute the two-dimensional slice matrices of the estimated expected signal $\widehat{\mathcal{X}}$:

   $$\widehat{\mathcal{X}}(:,:,i_3,\ldots,i_j,\ldots,i_N) = \mathbf{P}\mathcal{R}(:,:,i_3,\ldots,i_j,\ldots,i_N)\mathbf{Q}$$

2. output: estimated expected signal: $\widehat{\mathcal{X}}$.

Channel-by-channel SVD-based filtering is based on a common efficient method but exhibits a major drawback: it does not take into account the relationships between the components of the processed tensor. Moreover, channel-by-channel SVD-based filtering is appropriate only on some conditions. For example, applying SVD-based filtering to an image is generally appropriate when the rows or columns of an image are redundant, that is, linearly dependent. In this case, the rank K of the image is equal to the number of linearly independent rows or columns. It is only in this case that it would be safe to throw out eigenvectors from $K+1$ on. It is only in this special case that the noise subspace is orthogonal to the signal subspace. Otherwise, the noise simply increases the variance of the signal subspace and underestimating the signal subspace dimension would result in throwing out both signal and noise information. Thus, one would lose spatial resolution.

The next subsection presents a multiway filtering method that processes jointly, and not successively, each component of the data tensor.

3.2. Tensor filtering based on multimode PCA.

3.2.1. White decorrelated Gaussian noise and second-order-statistics-based method. Assuming that the dimension K_n of the signal subspace is known

for all $n = 1$ to N, one way to estimate the expected signal tensor $\mathcal{X}$ from the noisy data tensor $\mathcal{R} = \mathcal{X} + \mathcal{N}$ is to orthogonally project, for every nth-mode, the vectors of tensor $\mathcal{R}$ on the nth-mode signal subspace $E_1^{(n)}$ for all $n = 1$ to N. This statement is equivalent to replacing in (3.2) the filters $\mathbf{H}^{(n)}$ by the projectors $\mathbf{P}^{(n)}$ on the nth-mode signal subspace:

$$\widehat{\mathcal{X}} = \mathcal{R} \times_1 \mathbf{P}^{(1)} \times_2 \cdots \times_N \mathbf{P}^{(N)}. \tag{3.3}$$

In this last formulation, projectors $\mathbf{P}^{(n)}$ are estimated thanks to a multimode PCA applied to data tensor $\mathcal{R}$. This multimode PCA-based filtering generalizes the classical matrix filtering methods [16, 17, 21, 23, 24, 32] and implicitly supposes that the additive noise is *white* and *Gaussian*.

In the vector or matrix formulation, the definition of the projector on the signal subspace is based on the eigenvectors associated with the largest eigenvalues of the covariance matrix of the set of observation vectors. Hence, the determination of the signal subspace amounts to determine the best approximation (in the least-squares sense) of the observation matrix or the covariance matrix.

As an extension to the vector and matrix cases, in the tensor formulation, the projectors on the nth-mode vector spaces are determined by computing the rank-$(K_1, \ldots, K_N)$ approximation of $\mathcal{R}$ in the least-squares sense. From a mathematical point of view, the rank-$(K_1, \ldots, K_N)$ approximation of $\mathcal{R}$ is represented by tensor $\mathcal{R}^{K_1,\ldots,K_N}$ which minimizes the quadratic tensor Frobenius norm $\|\mathcal{R} - \mathcal{B}\|^2$ subject to the condition that $\mathcal{B} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ is a rank-$(K_1, \ldots, K_N)$ tensor. The description of the TUCKALS3 algorithm used in rank-$(K_1, \ldots, K_N)$ approximation is provided in the following.

Rank-$(K_1, \ldots, K_N)$ approximation - TUCKALS3 algorithm.

1. **Input:** data tensor $\mathcal{R}$, and dimensions $K_1, \ldots, K_N$ of all nth-mode signal subspaces.
2. **Initialization** $k = 0$: For $n = 1$ to N, calculate the projectors $\mathbf{P}_0^{(n)}$ given by HOSVD-$(K_1, \ldots, K_N)$:
   (a) nth-mode flatten $\mathcal{R}$ into matrix $\mathbf{R}_n$;
   (b) Compute the SVD of $\mathbf{R}_n$;
   (c) Compute matrix $\mathbf{U}_0^{(n)}$ formed by the K_n eigenvectors associated with the K_n largest singular values of $\mathbf{R}_n$. $\mathbf{U}_0^{(n)}$ is the initial matrix of the nth-mode signal subspace orthogonal basis vectors;
   (d) Form the initial orthogonal projector $\mathbf{P}_0^{(n)} = \mathbf{U}_0^{(n)}\mathbf{U}_0^{(n)^T}$ on the nth-mode signal subspace;
   (e) Compute the HOSVD-$(K_1, \ldots, K_N)$ of tensor $\mathcal{R}$ given by $\mathcal{B}_0 = \mathcal{R} \times_1 \mathbf{P}_0^{(1)} \times_2 \cdots \times_N \mathbf{P}_0^{(N)}$;
3. **ALS loop:**
   Repeat until convergence, that is, for example, while $\|\mathcal{B}_{k+1} - \mathcal{B}_k\|^2 > \epsilon$, $\epsilon > 0$ being a prior fixed threshold,
   (a) For $n = 1$ to N:
      i. Form $\mathcal{B}^{(n),k}$: $\mathcal{B}^{(n),k} = \mathcal{R} \times_1 \mathbf{P}_{k+1}^{(1)} \times_2 \cdots \times_{n-1} \mathbf{P}_{k+1}^{(n-1)} \times_{n+1} \mathbf{P}_k^{(n+1)} \times_{n+2} \cdots \times_N \mathbf{P}_k^{(N)}$;
      ii. nth-mode flatten tensor $\mathcal{B}^{(n),k}$ into matrix $\mathbf{B}_n^{(n),k}$;
      iii. Compute matrix $\mathbf{C}^{(n),k} = \mathbf{B}_n^{(n),k}\mathbf{R}_n^T$;
      iv. Compute matrix $\mathbf{U}_{k+1}^{(n)}$ composed of the K_n eigenvectors associated

with the K_n largest eigenvalues of $\mathbf{C}^{(n),k}$. $\mathbf{U}_k^{(n)}$ is the matrix of the nth-mode signal subspace orthogonal basis vectors at the k^{th} iteration;

v. Compute $\mathbf{P}_{k+1}^{(n)} = \mathbf{U}_{k+1}^{(n)} \mathbf{U}_{k+1}^{(n)^T}$;

(b) Compute $\mathcal{B}_{k+1} = \mathcal{R} \times_1 \mathbf{P}_{k+1}^{(1)} \times_2 \cdots \times_N \mathbf{P}_{k+1}^{(N)}$;

(c) Increment k.

4. **Output:** the estimated signal tensor is obtained through $\widehat{\mathcal{X}} = \mathcal{R} \times_1 \mathbf{P}_{k_{stop}}^{(1)} \times_2 \cdots \times_N \mathbf{P}_{k_{stop}}^{(N)}$. $\widehat{\mathcal{X}}$ is the rank-$(K_1, \ldots, K_N)$ approximation of $\mathcal{R}$, where k_{stop} is the index of the last iteration after the convergence of TUCKALS3 algorithm.

In this algorithm, the second order statistics come from the SVD of matrix $\mathbf{R}_n$ at step 2(b), which is equivalent, up to $\frac{1}{M_n}$ multiplicative factor, to the estimation of tensor $\mathcal{R}$ nth-mode vectors [39]. The definition of M_n is given in (2.1). In the same way, at step 3(a)iii, matrix $\mathbf{C}^{(n),k}$ is, up to $\frac{1}{M_n}$ multiplicative factor, the estimation of the covariance matrix between tensor $\mathcal{R}$ and tensor $\mathcal{B}^{(n),k}$ nth-mode vectors. According to step 3(a)ii, $\mathcal{B}^{(n),k}$ represents data tensor $\mathcal{R}$ filtered in every mth-mode but the nth-mode, by projection-filters $\mathbf{P}_l^{(m)}$, with $m \neq n$, $l = k$ if $m > n$ and $l = k + 1$ if $m < n$. TUCKALS3 algorithm has recently been used to process a multimode PCA in order to perform white noise removal in color images [36].

A good approximation of the rank-$(K_1, \ldots, K_N)$ approximation can simply be achieved by computing the HOSVD-$(K_1, \ldots, K_N)$ of tensor $\mathcal{R}$ [14, 34]. Indeed, the HOSVD-$(K_1, \ldots, K_N)$ of $\mathcal{R}$ consists of the initialization step of TUCKALS3 algorithm and hence can be considered as a suboptimal solution for the rank-$(K_1, \ldots, K_N)$ approximation of tensor $\mathcal{R}$ [14]. This HOSVD-based technique has recently been used in [39] for denoising and source separation of multicomponent seismic waves.

3.2.2. Correlated Gaussian noise and higher-order-statistics-based method. In practice, the condition of noise whiteness is not always fulfilled. Hence, in the case of an additive *correlated* Gaussian noise, the TUCKALS3 algorithm is theoretically incapable of providing a good estimation of the nth-mode signal subspaces since it is based on second order moments. A classical means to remove the Gaussian (noise) components is to use the higher order statistics, and especially the higher order cumulants. The tensor framework has been used to compute the fourth order cumulants as a means of solving the ICA problem [12]. Vasilescu and Terzopoulos introduced a multilinear ICA (N-mode ICA) for face recognition, which encodes the higher order statistics associated with each mode of the tensor [50, 51]. The related methods are based on the well-known cumulant property stating that the higher order cumulants of a Gaussian variable are null [31, 33].

As a consequence, in the case of an additive *correlated* Gaussian noise, a recent study [39] has proposed to improve the multimode PCA-based filtering by incorporating into the TUCKALS3 algorithm the fourth order cumulants instead of the second order moments.

From a practical point of view, second order matrices $\mathbf{C}^{(n),0}$ and $\mathbf{C}^{(n),k}$ at steps 2(b) and 3(a)iii of the TUCKALS3 algorithm are replaced with the corresponding fourth order cumulants. In the following, we present only the details of the procedure for matrix $\mathbf{C}^{(n),k}$. Obtaining the details concerning $\mathbf{C}^{(n),0}$ is straightforward.

We assume that $\{\mathbf{r}_p^{(n)}, p = 1, \ldots, M_n\}$ and $\{\mathbf{b}_p^{(n),k}, p = 1, \ldots, M_n\}$ are the M_n realizations of two random vectors $\mathbf{r}^{(n)}$ and $\mathbf{b}^{(n),k}$. In practice, we take as the realizations of these two random vectors the nth-mode vectors of data tensors $\mathcal{R}$ and $\mathcal{B}^{(n),k}$.

Matrix $\mathbf{C}^{(n),k}$ reads

$$\mathbf{C}^{(n),k} = \sum_{p=1}^{M_n} \mathbf{b}_p^{(n),k} \mathbf{r}_p^{(n)^T}. \tag{3.4}$$

The fourth order cumulants associated with vectors $\mathbf{r}^{(n)}$ and $\mathbf{b}^{(n),k}$ are denoted by

$$\mathcal{C}^{(n),k} = \mathrm{Cum}(\mathbf{b}^{(n),k}, \mathbf{b}^{(n),k^T}, \mathbf{r}^{(n)}, \mathbf{r}^{(n)^T}), \tag{3.5}$$

where $\mathrm{Cum}(\cdot)$ denotes the cumulant operator. $\mathcal{C}^{(n),k}$ is a fourth order super-symmetric tensor from $\mathbb{R}^{I_n \times I_n \times I_n \times I_n}$, whose generic term for indexes (i_1, i_2, j_1, j_2), for centered variables, is given by [19, 31]

$$\begin{aligned}(\mathcal{C}^{(n),k})_{i_1,i_2,j_1,j_2} &= \mathrm{E}[b_{i_1}^{(n),k} b_{i_2}^{(n),k} r_{j_1}^{(n)} r_{j_2}^{(n)}] \\ &\quad - \mathrm{E}[b_{i_1}^{(n),k} r_{j_1}^{(n)}]\mathrm{E}[b_{i_2}^{(n),k} r_{j_2}^{(n)}], -\mathrm{E}[b_{i_1}^{(n),k} r_{j_2}^{(n)}]\mathrm{E}[b_{i_2}^{(n),k} r_{j_1}^{(n)}]\end{aligned} \tag{3.6}$$

where $b_i^{(n),k}$ and $r_j^{(n)}$ are the ith and jth components of random vectors $\mathbf{b}^{(n),k}$ and $\mathbf{r}^{(n)}$, and $\mathrm{E}[\cdot]$ is the expectation operator. The practical estimation of $(\mathcal{C}^{(n),k})_{i_1,i_2,j_1,j_2}$ is given by

$$\begin{aligned}(\mathcal{C}^{(n),k})_{i_1,i_2,j_1,j_2} &= \frac{1}{M_n}\left(\sum_{p=1}^{M_n}\left(b_{i_1p}^{(n),k} b_{i_2p}^{(n),k} r_{j_1p}^{(n)} r_{j_2p}^{(n)}\right)\right) \\ &\quad - \frac{1}{{M_n}^2}\left(\sum_{p=1}^{M_n}\left(b_{i_1p}^{(n),k} r_{j_1p}^{(n)}\right)\right)\left(\sum_{p=1}^{M_n}\left(b_{i_2p}^{(n),k} r_{j_2p}^{(n)}\right)\right) \\ &\quad - \frac{1}{{M_n}^2}\left(\sum_{p=1}^{M_n}\left(b_{i_1p}^{(n),k} r_{j_2p}^{(n)}\right)\right)\left(\sum_{p=1}^{M_n}\left(b_{i_2p}^{(n),k} r_{j_1p}^{(n)}\right)\right).\end{aligned} \tag{3.7}$$

Here, $b_{ip}^{(n),k}$ and $r_{ip}^{(n)}$ are the elements at position (i, j) of tensors $\mathcal{B}^{(n),k}$ and $\mathcal{R}$ nth-mode flattening matrices $\mathbf{B}_n^{(n),k}$ and $\mathbf{R}_n$.

In the classical TUCKALS3 algorithm, the K_n nth-mode signal subspace basis vectors, given by matrix $\mathbf{U}^{(n),k}$, are estimated by computing, at step 3a, the eigenvectors associated with the K_n largest eigenvalues of matrix $\mathbf{C}^{(n),k}$. This amounts to computing the best lower rank-K_n approximation of $\mathbf{C}^{(n),k}$. In [41] fourth order cumulants are used instead of the covariance matrix because of their ability to remove Gaussian noise. Indeed, the fourth order cumulants of Gaussian variables are null. Therefore, when dealing with an additive *correlated* Gaussian noise, we also use fourth order cumulants [39].

The main drawback of fourth order cumulants is the high computational load to build every fourth order cumulant tensor associated with the nth-mode of the data tensor. This computational load depends on the size of the data tensor $\mathcal{R}$, that is, the values of I_n, for all $n = 1$ to N. One way to reduce the computational load has been proposed in [39] and consists in using the fourth order cumulant slice matrix. The cumulant slice matrix has initially been introduced in array processing for source localization or directions-of-arrival (DOA) estimation [7, 55, 54]. In [19, 55, 54], it is proved that the signal subspace spanned by the eigenvectors associated with the largest eigenvalues of a cumulant slice matrix is the same as signal subspace obtained from the whole cumulant tensor defined in (3.5) [55, 54]. Therefore, we use only the

eigenvectors of one cumulant slice matrix in our algorithm (see step 2(a)iii because the other cumulant slice matrices provide redundant information. The use of the fourth order cumulant slice matrix provides a much faster algorithm [54]. In our application, the fourth order cumulant slice matrix $\mathbf{C}_q^{(n),k}$ can be defined, from (3.6), by fixing the q^{th} component of vector $\mathbf{b}^{(n),k}$ as follows:

$$\left(\mathbf{C}_q^{(n),k}\right)_{ij} = \mathrm{E}\left[(b_q^{(n),k})^2 r_i^{(n)} r_j^{(n)}\right] - 2\mathrm{E}\left[b_q^{(n),k} r_i^{(n)}\right]\mathrm{E}\left[b_q^{(n),k} r_j^{(n)}\right]. \tag{3.8}$$

The practical estimation of $(\mathbf{C}_q^{(n),k})_{ij}$ can be given by

$$\left(\mathbf{C}_q^{(n),k}\right)_{ij} = \frac{1}{M_n}\left(\sum_{p=1}^{M_n}\left(b_{qp}^{(n),k}\right)^2 r_{ip}^{(n)} r_{jp}^{(n)}\right) - \frac{2}{M_n^2}\left(\sum_{p=1}^{M_n} b_{qp}^{(n),k} r_{ip}^{(n)}\right)\left(\sum_{p=1}^{M_n} b_{qp}^{(n),k} r_{jp}^{(n)}\right), \tag{3.9}$$

where $b_{ij}^{(n),k}$ and $r_{ij}^{(n)}$ are, respectively, the elements at position (i,j) in the nth-mode flattening matrices $\mathbf{B}_n^{(n),k}$ and $\mathbf{R}_n$ of tensors $\mathcal{B}^{(n),k}$ and $\mathcal{R}$.

As a consequence, in the case of an additive *correlated* Gaussian noise, the K_n nth-mode signal subspace basis vectors can now be estimated by computing matrix $\mathbf{C}_q^{(n),k}$ lower rank-K_n approximation. Then, the fourth order cumulant slice matrix-based multimode PCA-based filtering can be summarized as follows:

1. **Initialization** $k = 0$:
   For all $n = 1$ to N, $\mathbf{P}_0^{(n)} = \mathbf{U}_0^{(n)}\mathbf{U}_0^{(n)^T}$. $\mathbf{U}_0^{(n)}$ is the matrix of the K_n eigenvectors associated with the K_n largest eigenvalues of fourth order cumulant slice matrix $\mathbf{C}_q^{(n),0}$ of tensor $\mathcal{R}$ nth-mode vectors.
2. **ALS loop:**
   The steps (b) and (c) of the ALS loop are the same as in the algorithm "**rank-$(K_1,\ldots,K_N)$ approximation - TUCKALS3 algorithm**" described previously, and step (a) is replaced by
   (a) For $n = 1$ to N:
      i. $\mathcal{B}^{(n),k} = \mathcal{R}\times_1\mathbf{P}_{k+1}^{(1)}\times_2\cdots\times_{n-1}\mathbf{P}_{k+1}^{(n-1)}\times_{n+1}\mathbf{P}_k^{(n+1)}\times_{n+2}\cdots\times_N\mathbf{P}_k^{(N)}$;
      ii. Compute cumulant slice matrix $\mathbf{C}_q^{(n),k}$ associated with the fourth order cumulants of tensors $\mathcal{R}$ and $\mathcal{B}^{(n),k}$ nth-mode vectors. Every element of $\mathbf{C}_q^{(n),k}$ is given in (3.9);
      iii. Process matrix $\mathbf{C}_q^{(n),k}$ eigenvalue decomposition (EVD) and put the K_n eigenvectors associated with the K_n largest eigenvalues into $\mathbf{U}_{k+1}^{(n)}$;
      iv. Compute projector $\mathbf{P}_{k+1}^{(n)} = \mathbf{U}_{k+1}^{(n)}\mathbf{U}_{k+1}^{(n)^T}$;
3. **Output:** $\widehat{\mathcal{X}} = \mathcal{R}\times_1\mathbf{P}_{k_{stop}}^{(1)}\times_2\cdots\times_N\mathbf{P}_{k_{stop}}^{(N)}$, with k_{stop} being the index of the last iteration after convergence of the algorithm.

It was experimentally shown in [39] that when the parameter q involved in $\mathbf{C}_q^{(n),k}$ is chosen properly, multimode PCA filtering based on fourth order cumulants (denoted by rank$-\mathcal{C}(K_1,\ldots,K_N)$) and on fourth order cumulant slice matrix (denoted by rank$-\mathbf{C}_1(K_1,\ldots,K_N)$) give sensibly the same performances in regard to noise reduction in color images and multicomponent seismic waves.

3.3. Multiway Wiener filtering. Let $\mathbf{R}_n$, $\mathbf{X}_n$, and $\mathbf{N}_n$ be the nth-mode flattening matrices of tensors $\mathcal{R}$, $\mathcal{X}$, and $\mathcal{N}$, respectively.

In the previous subsection, the estimation of signal tensor $\mathcal{X}$ has been performed by projecting noisy data tensor $\mathcal{R}$ on each nth-mode signal subspace. The nth-mode

projectors have been estimated thanks to the use of multimode PCA achieved by rank-$(K_1, \ldots, K_N)$ approximation. Despite the good results given by this method, it is possible to improve the tensor filtering quality by determining nth-mode filters $\mathbf{H}^{(n)}$, $n = 1$ to N, in (3.2), which optimize an estimation criterion. The most classical method is to minimize the mean squared error between the expected signal tensor $\mathcal{X}$ and the estimated signal tensor $\widehat{\mathcal{X}}$ given in (3.2):

$$\mathrm{e}(\mathbf{H}^{(1)}, \ldots, \mathbf{H}^{(N)}) = \mathrm{E}[\|\mathcal{X} - \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \cdots \times_N \mathbf{H}^{(N)}\|^2]. \tag{3.10}$$

Due to the criterion which is minimized, filters $\mathbf{H}^{(n)}$, $n = 1$ to N, can be called "nth-mode Wiener filters" [38].

According to the calculations presented in Appendix A, especially from (A.1) to (A.15), the minimization of (3.10) with respect to filter $\mathbf{H}^{(n)}$, for fixed $\mathbf{H}^{(m)}$, $m \neq n$, leads to the following expression of nth-mode Wiener filter:

$$\mathbf{H}^{(n)} = \gamma_{\mathbf{XR}}^{(n)} {\mathbf{\Gamma}_{\mathbf{RR}}^{(n)}}^{-1}, \tag{3.11}$$

where

$$\gamma_{\mathbf{XR}}^{(n)} = \mathrm{E}\left[\mathbf{X}_n \mathbf{T}^{(n)} \mathbf{R}_n^T\right] \tag{3.12}$$

is the $\mathbf{T}^{(n)}$-weighted covariance matrix between the random column vectors of signal $\mathbf{X}_n$ and data $\mathbf{R}_n$, with

$$\mathbf{T}^{(n)} = \mathbf{H}^{(1)} \otimes \cdots \otimes \mathbf{H}^{(n-1)} \otimes \mathbf{H}^{(n+1)} \otimes \cdots \otimes \mathbf{H}^{(N)}, \tag{3.13}$$

where $\otimes$ stands for Kronecker product, and

$$\mathbf{\Gamma}_{\mathbf{RR}}^{(n)} = \mathrm{E}\left[\mathbf{R}_n \mathbf{Q}^{(n)} \mathbf{R}_n^T\right] \tag{3.14}$$

is the $\mathbf{Q}^{(n)}$-weighted covariance matrix of the data $\mathbf{R}_n$, with

$$\mathbf{Q}^{(n)} = {\mathbf{T}^{(n)}}^T \mathbf{T}^{(n)}. \tag{3.15}$$

In order to obtain $\mathbf{H}^{(n)}$ through (3.11), we suppose that the filters $\{\mathbf{H}^{(m)}, m = 1 \text{ to } N, m \neq n\}$ are known. Data tensor $\mathcal{R}$ is available, but signal tensor $\mathcal{X}$ is unknown. So, only the term $\mathbf{\Gamma}_{\mathbf{RR}}^{(n)}$ can be derived, and not the term $\gamma_{\mathbf{XR}}^{(n)}$. Hence, some more assumptions on $\mathcal{X}$ have to be made in order to overcome the indetermination over $\gamma_{\mathbf{XR}}^{(n)}$ [35, 38]. In the one-dimensional case, a classical assumption is to consider that a signal vector is a weighted combination of the signal subspace basis vectors. In extension to the tensor case, [35, 38] have proposed considering that the nth-mode flattening matrix $\mathbf{X}_n$ can be expressed as a weighted combination of K_n vectors from the nth-mode signal subspace $E_1^{(n)}$:

$$\mathbf{X}_n = \mathbf{V}_s^{(n)} \mathbf{O}^{(n)} \tag{3.16}$$

with $\mathbf{X}_n \in \mathbb{R}^{I_n \times M_n}$, and $\mathbf{V}_s^{(n)} \in \mathbb{R}^{I_n \times K_n}$ being the matrix containing the K_n orthonormal basis vectors of nth-mode signal subspace $E_1^{(n)}$. Matrix $\mathbf{O}^{(n)} \in \mathbb{R}^{K_n \times M_n}$ is a weight matrix and contains the whole information on expected signal tensor $\mathcal{X}$. This model implies that signal nth-mode flattening matrix $\mathbf{X}_n$ is orthogonal to nth-mode

noise flattening matrix $\mathbf{N}_n$, since signal subspace $E_1^{(n)}$ and noise subspace $E_2^{(n)}$ are supposed mutually orthogonal.

Supposing that noise $\mathcal{N}$ in (3.1) is white, Gaussian, and independent from signal $\mathcal{X}$, and introducing the signal model (3.16) in (3.11) leads to a computable expression of nth-mode Wiener filter $\mathbf{H}^{(n)}$ (see Appendix B),

$$\mathbf{H}^{(n)} = \mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \mathbf{\Lambda}_{\mathbf{\Gamma s}}^{(n)^{-1}} \mathbf{V}_s^{(n)^T}, \tag{3.17}$$

where $\gamma_{\mathbf{OO}}^{(n)} \mathbf{\Lambda}_{\mathbf{\Gamma s}}^{(n)^{-1}}$ is a diagonal weight matrix given by

$$\gamma_{\mathbf{OO}}^{(n)} \mathbf{\Lambda}_{\mathbf{\Gamma s}}^{(n)^{-1}} = \operatorname{diag}\left[\frac{\beta_1}{\lambda_1^{\Gamma}}, \cdots, \frac{\beta_{K_n}}{\lambda_{K_n}^{\Gamma}}\right], \tag{3.18}$$

where $\lambda_1^{\Gamma}, \ldots, \lambda_{K_n}^{\Gamma}$ are the K_n largest eigenvalues of $\mathbf{Q}^{(n)}$-weighted covariance matrix $\mathbf{\Gamma}_{\mathbf{RR}}^{(n)}$ (see (3.14)). Parameters $\beta_1, \ldots, \beta_{K_n}$ depend on $\lambda_1^{\gamma}, \ldots, \lambda_{K_n}^{\gamma}$, which are the K_n largest eigenvalues of $\mathbf{T}^{(n)}$-weighted covariance matrix $\gamma_{\mathbf{RR}}^{(n)} = \mathrm{E}[\mathbf{R}_n \mathbf{T}^{(n)} \mathbf{R}_n^T]$, according to the following relation:

$$\beta_{k_n} = \lambda_{k_n}^{\gamma} - \sigma_{\Gamma}^{(n)^2} \ \forall\ k_n = 1, \ldots, K_n. \tag{3.19}$$

Superscript γ refers to the $\mathbf{T}^{(n)}$-weighted covariance and subscript $\mathbf{\Gamma}$ to the $\mathbf{Q}^{(n)}$-weighted covariance. $\sigma_{\Gamma}^{(n)^2}$ is the degenerated eigenvalue of noise $\mathbf{T}^{(n)}$-weighted covariance matrix $\gamma_{\mathbf{NN}}^{(n)} = \mathrm{E}\left[\mathbf{N}_n \mathbf{T}^{(n)} \mathbf{N}_n^T\right]$. Thanks to the additive noise and the signal independence assumptions, the $I_n - K_n$ smallest eigenvalues of $\gamma_{\mathbf{RR}}^{(n)}$ are equal to $\sigma_{\Gamma}^{(n)^2}$ and thus can be estimated by the following relation:

$$\widehat{\sigma}_{\Gamma}^{(n)^2} = \frac{1}{I_n - K_n} \sum_{k_n = K_n + 1}^{I_n} \lambda_{k_n}^{\gamma}. \tag{3.20}$$

In order to determine the nth-mode Wiener filters $\mathbf{H}^{(n)}$ that minimize the mean squared error (3.10), the alternating least squares (ALS) algorithm has been proposed in [35, 38]. It can be summarized in the following steps:

1. **Initialization** $k = 0$**:** $\mathcal{R}^0 = \mathcal{R} \Leftrightarrow \mathbf{H}_0^{(n)} = \mathbf{I}_{In}$, Identity matrix, $\forall\ n = 1 \ldots N$.
2. **ALS loop:**
   Repeat until convergence, that is, $\left\|\mathcal{R}^{k+1} - \mathcal{R}^k\right\|^2 < \epsilon$, with $\epsilon > 0$ prior fixed threshold,
   (a) for $n = 1$ to N:
      i. Form $\mathcal{R}^{(n),k}$: $\mathcal{R}^{(n),k} = \mathcal{R} \times_1 \mathbf{H}_{k+1}^{(1)} \times_2 \cdots \times_{n-1} \mathbf{H}_{k+1}^{(n-1)} \times_{n+1} \mathbf{H}_k^{(n+1)} \times_{n+2} \ldots \times_N \mathbf{H}_k^{(N)}$;
      ii. Determine $\mathbf{H}_{k+1}^{(n)} = \arg\min_{\mathbf{Z}^{(n)}} \left\|\mathcal{X} - \mathcal{R}^{(n),k} \times_n \mathbf{Z}^{(n)}\right\|^2$
      subject to $\mathbf{Z}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ thanks to the following procedure:
         A. nth-mode flatten $\mathcal{R}^{(n),k}$ into $\mathbf{R}_n^{(n),k} = \mathbf{R}_n(\mathbf{H}_{k+1}^{(1)} \otimes \cdots \otimes \mathbf{H}_{k+1}^{(n-1)} \otimes \mathbf{H}_k^{(n+1)} \otimes \cdots \otimes \mathbf{H}_k^{(N)})^T$, and $\mathcal{R}$ into $\mathbf{R}_n$;
         B. Compute $\gamma_{\mathbf{RR}}^{(n)} = \mathrm{E}[\mathbf{R}_n \mathbf{R}_n^{(n),k^T}]$,
         C. Determine $\lambda_1^{\gamma}, \ldots, \lambda_{K_n}^{\gamma}$, the K_n largest eigenvalues of $\gamma_{\mathbf{RR}}^{(n)}$;

D. For $k_n = 1$ to I_n, estimate $\sigma_\Gamma^{(n)^2}$ thanks to (3.20) and for $k_n = 1$ to K_n, estimate β_{k_n} thanks to (3.19);
E. Compute $\mathbf{\Gamma}_{\mathbf{RR}}^{(n)} = \mathrm{E}[\mathbf{R}_n^{(n),k}\mathbf{R}_n^{(n),k^T}]$;
F. Determine $\lambda_1^\Gamma, \ldots, \lambda_{K_n}^\Gamma$, the K_n largest eigenvalues of $\mathbf{\Gamma}_{\mathbf{RR}}^{(n)}$;
G. Determine $\mathbf{V}_s^{(n)}$, the matrix of the K_n eigenvectors associated with the K_n largest eigenvalues of $\mathbf{\Gamma}_{\mathbf{RR}}^{(n)}$;
H. Compute the weight matrix $\gamma_{\mathbf{OO}}^{(n)}\mathbf{\Lambda}_{\mathbf{\Gamma s}}^{(n)^{-1}}$ given in (3.18);
I. Compute $\mathbf{H}_{k+1}^{(n)}$, the nth-mode Wiener filter at the $(k+1)^{\text{th}}$ iteration, using (3.17);

(b) Form $\mathcal{R}^{k+1} = \mathcal{R} \times_1 \mathbf{H}_{k+1}^{(1)} \times_2 \cdots \times_N \mathbf{H}_{k+1}^{(N)}$;
(c) Increment k;

3. **output:** $\widehat{\mathcal{X}} = \mathcal{R} \times_1 \mathbf{H}_{k_{stop}}^{(1)} \times_2 \cdots \times_N \mathbf{H}_{k_{stop}}^{(N)}$, with k_{stop} being the last iteration after convergence of the algorithm.

In subsection 3.2, we presented the adaptation of multimode PCA to the case of a noncorrelated Gaussian noise, by using higher order statistics. In the same way, it is possible to use higher order statistics for multiway Wiener filtering. For this, one should replace step 2(a)iiB by step 2(a)ii of the ALS loop in subsection 3.2, and replace step 2(a)iiE by the computation of the cumulant slice $\mathbf{C}_q^{(n),k}$ associated with the fourth order cumulants of matrix $\mathbf{R}_n^{(n),k}$ and matrix $(\mathbf{R}_n^{(n),k})^T$. Elements of $\mathbf{C}_q^{(n),k}$ are given in (3.9).

4. Simulation results. In the following simulations, the channel-by-channel SVD-based filtering defined in subsection 3.1 and the rank-$(K_1, \ldots, K_N)$ approximation-based multiway and multidimensional filtering are applied to the denoising of color images and multispectral images and to the denoising of seismic signals. Color images, multispectral images, and seismic signals can be represented by a third order tensor from $\mathbb{R}^{I_1 \times I_2 \times I_3}$, where I_1, I_2, and I_3 take different values. In all these applications, the efficiency of denoising is tested in the presence of an additive Gaussian noise, either correlated or not.

A multidimensional and multiway white Gaussian noise $\mathcal{N}$ which is added to signal tensor $\mathcal{X}$ can be expressed as

$$\mathcal{N} = \alpha \cdot \mathcal{G}, \tag{4.1}$$

where every element of $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is an independent realization of a normalized centered Gaussian law and where α is a coefficient that permits to set the signal-to-noise ratio (SNR) in noisy data tensor $\mathcal{R}$.

When we process images impaired by correlated Gaussian noise, the noise which is added is a third order tensor defined by

$$\mathcal{N}^c = \mathcal{N} \times_1 \mathbf{W}^{(1)} \times_2 \mathbf{W}^{(2)} \times_3 \mathbf{W}^{(3)}, \tag{4.2}$$

where every element of $\mathcal{N}$ represents an independent realization of a white Gaussian noise and $\mathbf{W}^{(n)}$ is a weight matrix in the nth-mode, $n = 1, 2, 3$.

In order to evaluate the performances of the overviewed tensor signal processing methods, a particular performance criterion is employed as proposed in [38, 39].

4.1. Performance criterion. Following the representation of (3.1), the multiway noisy data tensor is expressed as $\mathcal{R} = \mathcal{X} + \mathcal{N}$, where $\mathcal{X}$ is the expected signal

tensor and $\mathcal{N}$ is the additive noise tensor. Let us define the SNR, in dB, in the noisy data tensor by

$$\text{SNR} = 10 \log \left(\frac{\|\mathcal{X}\|^2}{\|\mathcal{N}\|^2} \right). \tag{4.3}$$

In order to a posteriori verify the quality of the estimated signal tensor, we use the normalized quadratic error criterion (NQE) defined as follows:

$$\text{NQE}(\widehat{\mathcal{X}}) = \frac{\|\widehat{\mathcal{X}} - \mathcal{X}\|^2}{\|\mathcal{X}\|^2}. \tag{4.4}$$

The NQE criterion permits a quantitative comparison of the channel-by-channel SVD-based filtering and the rank-(K_1, K_2, K_3) approximation multiway and multidimensional filtering. Considering this criterion, we expect the rank-(K_1, K_2, K_3) approximation to give better results than the channel-by-channel SVD-based filtering method.

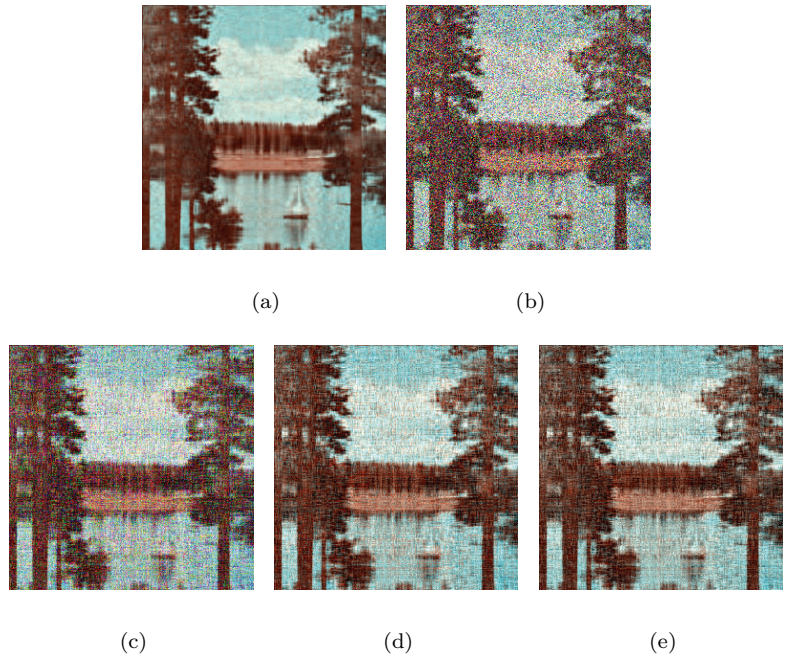
4.2. Denoising of color images. Denoising of color images has been studied in several works [6, 40, 43]. Some solutions have been brought from the field of wavelet processing, exhibiting good results in terms of output SNR. These studies concern only bidimensional data, whereas the methods that we compare are adapted to the processing of third order tensors as a whole, and in particular to three-channel images. We focus on subspace-based methods. We first consider the channel-by-channel SVD-based filtering, the rank-(K_1, K_2, K_3) approximation and multiway Wiener filtering (Wmm-(K_1, K_2, K_3)), applied to images impaired by an additive white Gaussian noise.

Then we present the results obtained with rank-(K_1, K_2, K_3) based on second order and higher order statistics, applied to images impaired by an additive correlated Gaussian noise. We compare the performances of the methods applied in this subsection in terms of denoising efficiency and computational load.

4.2.1. Denoising of a color image impaired by additive Gaussian noise. Let us consider the "sailboat" standard color image of Figure 4.1(a) represented as a third order tensor $\mathcal{X} \in \mathbb{R}^{256 \times 256 \times 3}$. The ranks of the signal subspace for each mode are 30 for the 1st-mode, 30 for the 2nd-mode, and 2 for the 3rd-mode. This is fixed thanks to the following process. For Figure 4.1(a), we took the standard nonnoisy sailboat image and artificially reduced the ranks of the nonnoisy image, that is, we set the parameters (K_1, K_2, K_3) to $(30, 30, 2)$, thanks to the truncation of HOSVD. This ensures that, for each mode, the rank of the signal subspace is lower than the corresponding dimension. This also permits us to evaluate the performances of the filtering methods applied, independently from the accuracy of the estimation of the values of the ranks by MDL or AIC criterion.

Figure 4.1(b) shows the noisy image resulting from the impairment of Figure 4.1(a) and represented as $\mathcal{R} = \mathcal{X} + \mathcal{N}$. Third-order noise tensor $\mathcal{N}$ is defined by relation (4.1) by choosing α such that, considering previous definition of (4.3), the SNR in the noisy image of Figure 4.1(b) is 8.1 dB. In these simulations, the value of the parameter K of channel-by-channel SVD-based filtering, the values of the dimensions of the row and column signal subspace are supposed to be known and fixed to 30. In the same way, parameters (K_1, K_2, K_3) of rank-(K_1, K_2, K_3) approximation are fixed to $(30, 30, 2)$.

The channel-by-channel SVD-based filtering of noisy image $\mathcal{R}$ (see Figure 4.1(b)) yields the image of Figure 4.1(c), and rank-$(30, 30, 2)$ approximation of noisy data ten-



(a) (b)

(c) (d) (e)

FIG. 4.1. (a) *Nonnoisy image.* (b) *Image to be processed, impaired by an additive white Gaussian noise, with SNR* $= 8.1$ *dB.* (c) *Channel-by-channel SVD-based filtering of parameter* $K = 30$. (d) *rank-(*$30, 30, 2$*) approximation.* (e) *Wmm-(*$30, 30, 2$*) filtering.*

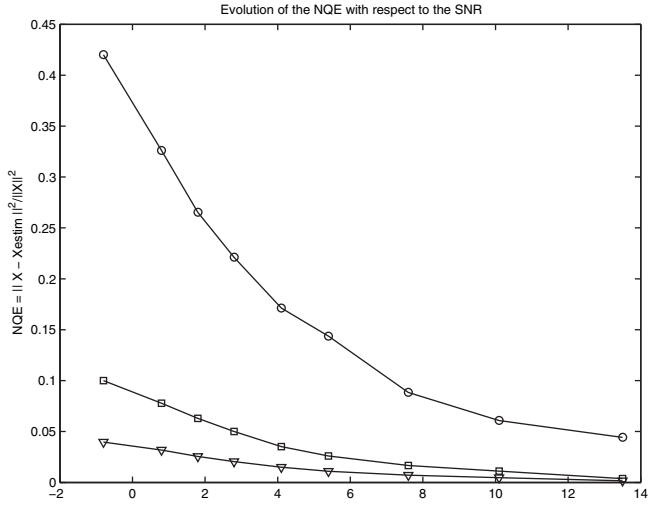


FIG. 4.2. *NQE evolution with respect to SNR* (dB): *channel-by-channel SVD-based filtering of parameter* 30 *(-○-), rank-(*$30, 30, 2$*) approximation (-□-), Wmm-(*$30, 30, 2$*) filtering (-▽-).*

sor $\mathcal{R}$ yields the image of Figure 4.1(d). The NQE, defined in (4.4), permits a qualitative comparison between channel-by-channel SVD-based filtering and rank-$(30, 30, 2)$ approximation. Figure 4.2, which presents the evolution of the NQE with respect to SNR varying from 3 dB to 18 dB, shows the NQE obtained with Wmm-$(30, 30, 2)$ is lower than the NQE obtained with the filtering with rank-$(30, 30, 2)$ approximation.

For this simulation, the rank-(K_1, K_2, K_3) approximation gives better results than channel-by-channel SVD-based filtering according to the NQE criterion. From the resulting image, presented on Figure 4.1(d), we note that dimension reduction leads to a loss of spatial resolution. However, the choice of a set of values K_1, K_2, K_3 which are small enough is the condition for an efficient noise reduction effect.

Therefore, a trade-off should be considered between noise reduction and detail preservation. This trade-off was discussed in [42]. We were interested in using the minimum description length (MDL) criterion [53], applied to the left singular values of the flattening matrices computed over the successive nth-modes. As a rule of thumb, the MDL criterion overestimates the value of parameters K_1, K_2, and K_3. This results in the preservation of the details in the processed image, at the expense of an efficient denoising.

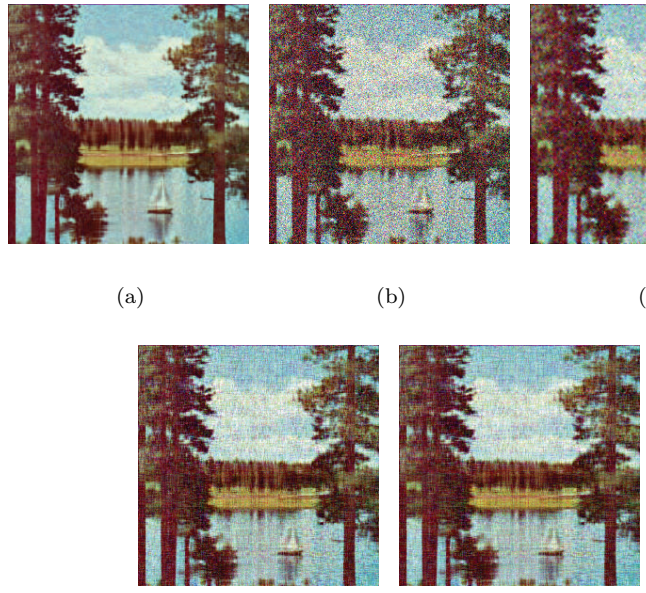
Concerning the qualitative results obtained with this color image, we note that the intraclass variance of the pixel values of each component (or color mode) of the resulting image is lower for the image obtained with Wmm-$(30, 30, 2)$ than for those images obtained with other methods applied in this subsection. This allows, for example, appling after denoising a high level classification method with a higher efficiency than when classification is applied after channel-by-channel SVD-based filtering or HOSVD-$(30, 30, 2)$.

For the $256 \times 256 \times 3$ sailboat image of Figure 4.1, the computational times needed when Matlab programs are used on a 3 Ghz Pentium 4 processor running Windows are as follows. HOSVD-$(30, 30, 2)$ lasts 1.61 seconds, the channel-by-channel SVD-based filtering lasts 1.94 seconds, the rank-$(30, 30, 2)$ approximation run with 25 iterations lasts 54.1 seconds, and Wmm-$(30, 30, 2)$ run with 25 iterations lasts 40.0 seconds.

The results presented in Figure 4.1 show that Wmm-(K_1, K_2, K_3) allows one to obtain better results in terms of NQE with a computational load which is lower than that of the rank-(K_1, K_2, K_3) approximation. In the next two examples we study the influence of the values of the nth-mode ranks. In the example of Figure 4.3 we set, in the same way as in the previous example, the ranks of the truncated image to $(30, 30, 3)$ (see Figure 4.3(a)). Note that $K_3 = I_3 = 3$. Thus the assumption $K_3 < I_3$ is not fulfilled. We aim at studying the behavior of the proposed tensor filtering algorithms when the color mode rank is equal to the color mode dimension ($K_3 = I_3$). The truncated image is impaired by a noncorrelated Gaussian noise such that SNR = 8.1 dB (see Figure 4.3(b)). The results obtained show that channel-by-channel Wiener-based filtering of parameter $K = 30$ (see Figure 4.3(c)) is outperformed by rank-$(30, 30, 3)$ approximation (see Figure 4.3(d)) and Wmm-$(30, 30, 3)$ (see Figure 4.3(e)). Indeed, the proposed tensor filtering algorithms rely on an ALS loop which permits us to take into account the relationships between the filters of each mode when multiway filters are used. In particular, concerning multiway Wiener filtering, it can be adapted to the case where it is applied with $K_3 = I_3$. For this, the weight matrix $\gamma^{(3)}_{\mathbf{OO}} \mathbf{\Lambda}^{(3)^{-1}}_{\mathbf{\Gamma s}}$ of step 2aiiH of the multiway Wiener filtering algorithm presented in subsection 3.3 is set to identity. That is, $\mathbf{H}^{(3)}$ is replaced by $\mathbf{P}^{(3)}$. We adapted the algorithm in order to take into account the channel mode information for the computation of the two spatial filters thanks to the ALS loop.

This proves the interest of multiway filtering even in the case where the rank of the signal subspace along the third mode is equal to the number of channels.

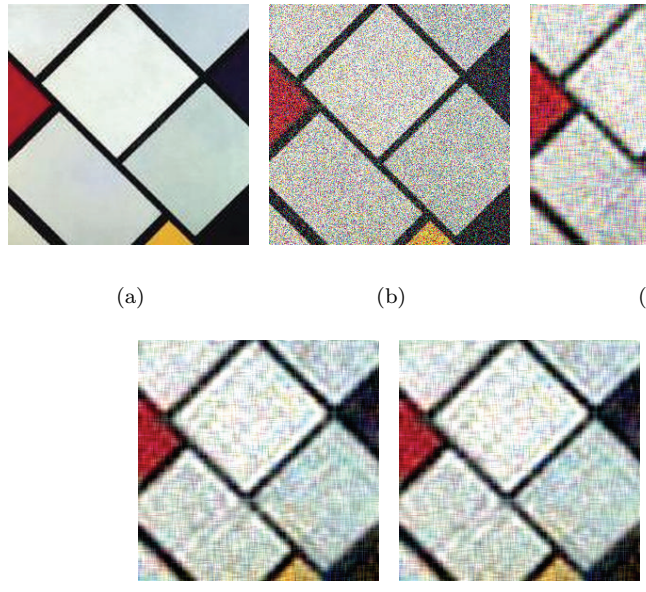
In the example of Figure 4.4 we study the case where the ranks of the signal subspaces are underestimated for the spatial modes. Let us consider the "Mondriaan" standard color image of Figure 4.4 represented as a third order tensor $\mathcal{X} \in \mathbb{R}^{256 \times 256 \times 3}$. We set the ranks of the truncated image to $(150, 150, 3)$. The ranks along the spa-



(a) (b) (c)

(d) (e)

FIG. 4.3. (a) *Nonnoisy image.* (b) *Image to be processed, impaired by an additive white Gaussian noise, with SNR* $= 8.1$ *dB.* (c) *Channel-by-channel Wiener-based filtering of parameter* $K = 30$. (d) *rank-(*$30, 30, 3$*) approximation.* (e) *Wmm-(*$30, 30, 3$*) filtering.*



(a) (b) (c)

(d) (e)

FIG. 4.4. (a) *Nonnoisy image.* (b) *Image to be processed, impaired by an additive white Gaussian noise, with SNR*$= 8.0$ *dB.* (c) *Channel-by-channel SVD-based filtering of parameter* $K = 19$. (d) *rank-(*$19, 19, 3$*) approximation.* (e) *Wmm-(*$19, 19, 3$*) filtering.*

tial modes will be fixed intentionally to a value which is smaller than 150 when the reviewed methods are applied. Figure 4.4(a) gives the nonnoisy image, Figure 4.4(b) shows the noisy image resulting from the impairment, with SNR= 8.0 dB, of the image of Figure 4.4(a). Figure 4.4(c) gives the result obtained with channel-by-channel SVD-based filtering of parameter $K = 19$. Figure 4.4(d) gives the result obtained with rank-$(19, 19, 3)$ approximation, and Figure 4.4(e) gives the result obtained with Wmm-$(19, 19, 3)$ filtering. Note that choosing (K_1, K_2, K_3)=$(19, 19, 3)$ results in throwing out both signal and noise information along the spatial modes, as the ranks of the noisy image are $(150, 150, 3)$. Underestimating the ranks along the spatial modes induces some blurry effect in the result images: part of the spatial resolution is lost. The presented subspace-based algorithms perform well if there is a high level of redundancy in the column or row space or if the image exhibits many soft or blurry edges, and the nth-mode ranks are not underestimated.

4.2.2. HOSVD-(K_1, K_2, K_3), rank-(K_1, K_2, K_3) approximation based on second order and higher order statistics, applied to an image impaired by an additive correlated Gaussian noise. The purpose here is to compare methods based on second order statistics with methods based on higher order statistics when an image is impaired by a correlated Gaussian noise. Figure 4.5 shows the results obtained with the HOSVD-(K_1, K_2, K_3), and the rank-(K_1, K_2, K_3) approximation based on second order and higher order statistics, used for the denoising of an image impaired by an additive correlated Gaussian noise. We consider the nonnoisy image of Figure 4.5(a) whose ranks are fixed to $(30, 30, 2)$: we artificially reduced the ranks of the nonnoisy image, that is, we set the parameters (K_1, K_2, K_3) to (30,30,2), thanks to the truncation of HOSVD. This image is impaired by a correlated Gaussian noise (see (4.2)). Figure 4.5(b) shows the noisy image. The result of HOSVD-(K_1, K_2, K_3) is given in Figure 4.5(c), and the result of rank-(K_1, K_2, K_3) approximation based on second order statistics is given in Figure 4.5(d), the result of

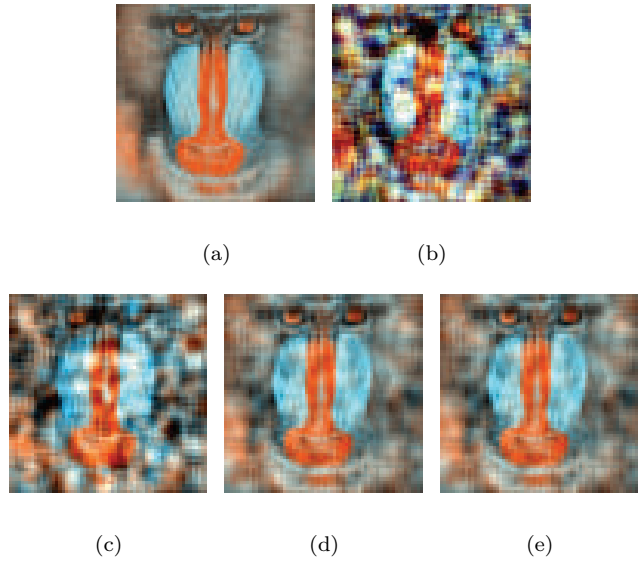


FIG. 4.5. (a) *Initial nonnoisy image.* (b) *Initial image with an additive correlated Gaussian noise, SNR = 2.48 dB.* (c) *HOSVD-(30, 30, 2) .* (d) *rank-$\mathcal{C}$(30, 30, 2) approximation.* (e) *rank-$\mathbf{C}_1$(30, 30, 2) approximation.*

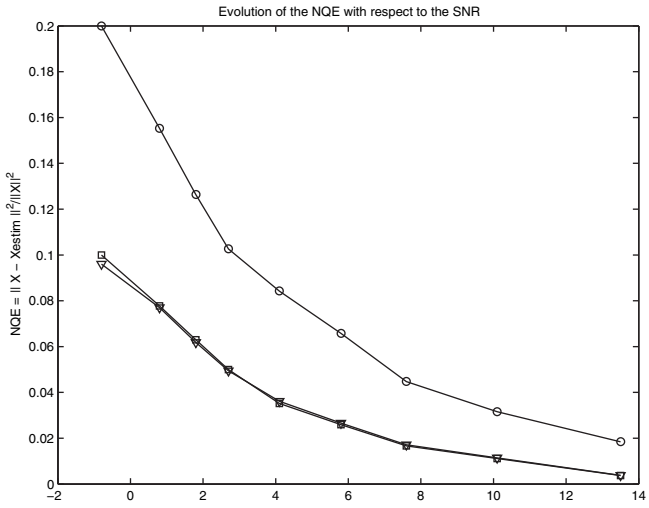


FIG. 4.6. *Evolution of the NQE with respect to the SNR(dB) for each tensor filtering method:* ○*: HOSVD-*$(30, 30, 2)$*;* ▽*: rank-*$\mathcal{C}(30, 30, 2)$*;* □*: rank-*$\mathbf{C}_1(30, 30, 2)$*.*

rank$-\mathcal{C}(K_1, K_2, K_3)$ approximation based on higher order statistics is given in Figure 4.5(e). The evolution of the NQE with respect to the SNR for HOSVD-$(30, 30, 2)$, rank-$\mathcal{C}(30, 30, 2)$ approximation based on fourth order cumulants, rank-$\mathbf{C}_1(30, 30, 2)$ approximation based on one slice of the fourth order cumulants is represented in Figure 4.6.

The main conclusion from Figure 4.5 is that the methods based on fourth order cumulants give similar visual results and better results than HOSVD-$(30, 30, 2)$. Whatever the SNR, the methods based on fourth order cumulants give a lower NQE value than the methods based on second order statistics. The method based on fourth order cumulant slice matrix gives sensibly the same NQE values as the method based on fourth order cumulants.

For the $256 \times 256 \times 3$ baboon image of Figure 4.5, the computational times needed in the same conditions of processor and software as in previous subsection are the following: HOSVD-$(30, 30, 2)$ lasts 1.61 seconds, rank-$\mathcal{C}(30, 30, 2)$ based filtering lasts 2h. 11 min. 40 seconds, rank-$\mathbf{C}_1(30, 30, 2)$ lasts 3 min. 50 seconds.

4.3. Denoising of multispectral images. The results obtained from the processing of a multispectral image composed of 72 rows, 160 columns and 100 spectral channels representing a truck are considered. This set of spectral images can be represented as a tensor $\mathcal{X} \in \mathbb{R}^{72 \times 160 \times 100}$. Images shown on Figures 4.7(a) to 4.7(e) represent channels 30 to 34 of the multispectral image. To evaluate the performances of the reviewed methods, some signal-independent white Gaussian noise $\mathcal{N}$ is added to $\mathcal{X}$ and results in noisy tensor $\mathcal{R} = \mathcal{X} + \mathcal{N}$. Channels 30 to 34 of noisy multispectral image represented as $\mathcal{R}$ are shown in Figures 4.7(f) to 4.7(j), and correspond to a noise impairment level SNR $= -1$ dB. Figures 4.7(k) to 4.7(o) represent channels 30 to 34 of the multispectral image obtained by applying channel-by-channel-based SVD-filtering to noisy image $\mathcal{R}$. Finally, Figures 4.7(p) to 4.7(t) represent channels 30 to 34 of the multispectral image obtained after applying rank-(30,30,30) approximation to noisy image $\mathcal{R}$. This last simulation clearly shows that the rank-(30,30,30) approximation-based filtering gives better results than channel-by-channel SVD-based

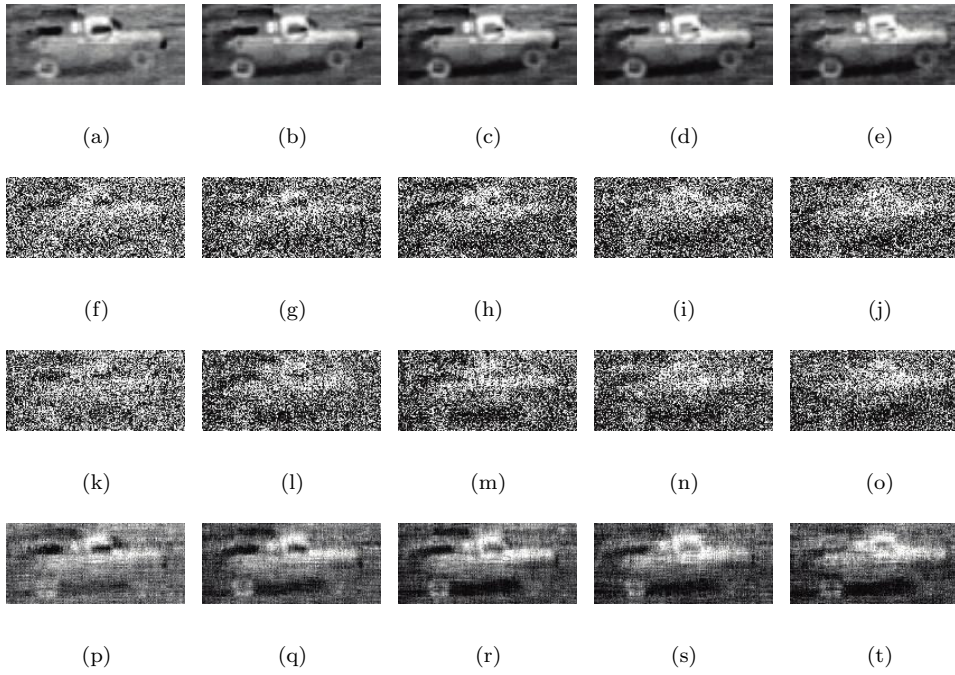


FIG. 4.7. *Channels* 30 *to* 34 *of the processed multispectral images are presented:* (a)–(e) *Nonnoisy multispectral image.* (f)–(j) *Impaired multispectral image.* (k)–(o) *Results obtained with channel-by-channel SVD filtering.* (p)–(t) *Results obtained with rank-*$(30, 30, 30)$ *approximation.*

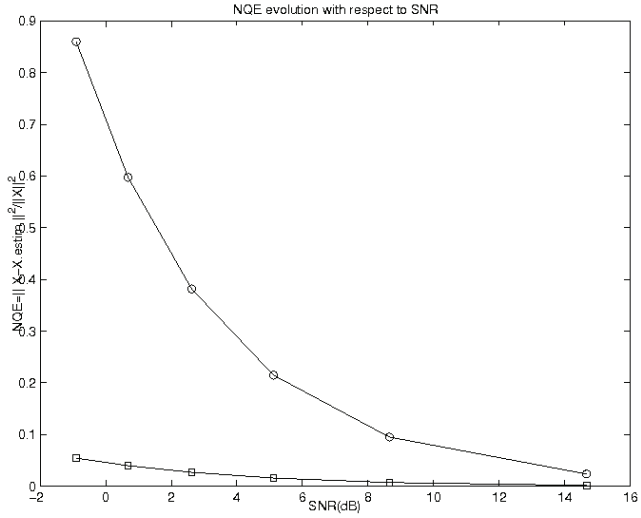


FIG. 4.8. *NQE evolution with respect to SNR (from* -1 *to* 15 *dB): channel-by-channel SVD-based filtering of parameter* 30 *(-∘-), and rank-(*$30, 30, 30$*) approximation (-□-).*

filtering in regard to denoising. Moreover, the evolution of the NQE with respect to the SNR varying from -1 dB to 15 dB, represented in Figure 4.8, shows that the NQE obtained with Wmm-(K_1, K_2, K_3) is lower than the NQE obtained with a previously existing method.

For this simulation the estimation quality, respect to the NQE criterion, is better for rank-(K_1, K_2, K_3) approximation, compared to channel-by-channel SVD-based filtering. Superiority of rank-(K_1, K_2, K_3) approximation compared to channel-by-channel SVD-based filtering is confirmed.

According to the simulations performed on a color image and on a multispectral image, it is possible to conclude that the more channels the image is composed of, the better the denoising. This can be explained by a better estimation of projectors on 1st-mode and 2nd-mode signal subspaces in the case of the multispectral image. Indeed, the number of spectral channels in a multispectral image is much larger than in a color image. Equivalently, I_3 is much larger than 3, so M_1 and M_2 are much larger than for a color image, and the estimation of matrices $\mathbf{C}^{(1),k}$ and $\mathbf{C}^{(2),k}$ presented in (3.4) are computed with more realization vectors.

4.4. Statistical performances. The goal of the following simulation is to test the robustness to noise of channel-by-channel SVD-based filtering of parameter K and of rank-(K_1, K_2, K_3) approximation, with respect to the NQE criterion. We process the Sailboat standard color image, impaired by an additive Gaussian noise, with SNR values varying from -0.7 dB to 15 dB; 100 trials are performed. For each trial one realization of additive Gaussian noise is simulated and added to the nonnoisy image. The mean and standard deviation are computed over the NQE values obtained each time the channel-by-channel SVD-based filtering and the rank-(K_1, K_2, K_3) approximation are run. The evolution of the mean NQE

$$m_{\mathrm{NQE}} = \frac{1}{100} \sum_{i=1}^{100} \mathrm{NQE}_i, \tag{4.5}$$

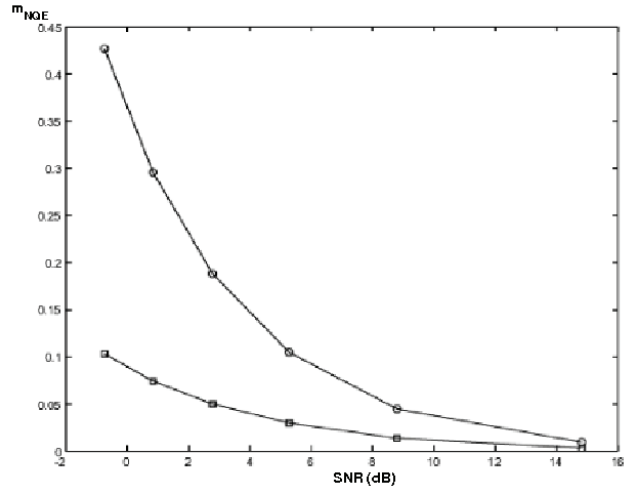
where index i refers to the ith noise realization, is represented in Figure 4.9(a) with respect to SNR. The evolution of the standard deviation of the NQE,

$$\mathrm{std}_{\mathrm{NQE}} = \sqrt{\frac{1}{100} \sum_{i=1}^{100} (\mathrm{NQE}_i - m_{\mathrm{NQE}})^2}, \tag{4.6}$$

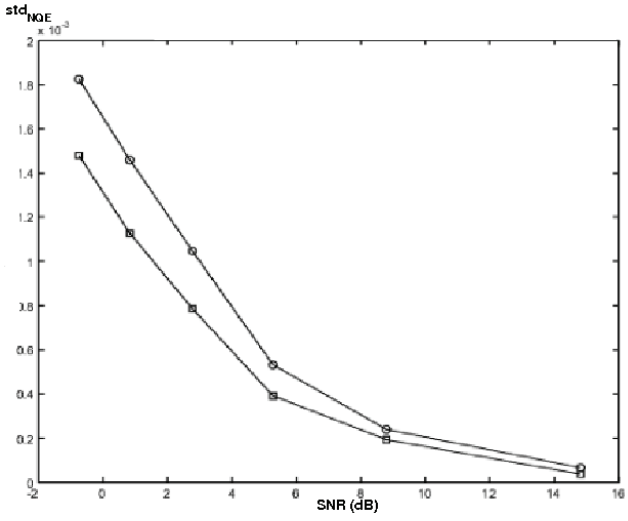
is represented in Figure 4.9(b), with respect to the SNR. Figure 4.9 shows that the mean and standard deviation values of the NQE obtained with rank-(K_1, K_2, K_3) approximation and computed over 100 noise realizations are both lower than the mean and the standard deviation values obtained with channel-by-channel SVD-based filtering. Thus, for these simulations, the rank-(K_1, K_2, K_3) approximation gives better results than channel-by-channel SVD-based filtering in regard to the robustness of tensor estimation and considering the NQE criterion.

4.5. Filtering of a multicomponent seismic type signal.

4.5.1. Filtering of a multicomponent seismic type signal impaired by an additive white Gaussian noise. In this simulation, a multicomponent seismic wave is received on a linear antenna composed of 10 sensors. The direction of propagation of the wave is assumed to be contained in a plane which is orthogonal to the antenna. The three components of the wave, represented as signal tensor $\mathcal{X}$, are called Component 1, Component 2, and Component 3 and are represented in Figures 4.10(a)–(c). In each seismic slice, the x-axis corresponds to the time sampling (200 or 100 time samples) and the y-axis corresponds to the spatial sensors (10 sensors). Each consecutive component presents a $\frac{\pi}{2}$ radian phase shift. The three components of noisy data tensor



(a)



(b)

FIG. 4.9. *(∘): results obtained with channel-by-channel SVD-based filtering of parameter* 30*; (□): results obtained with rank-(*30, 30, 2*) approximation.* (a) *Evolution of the mean NQE with respect to SNR (dB).* (b) *Evolution of the standard deviation of NQE with respect to SNR (dB).*

$\mathcal{R}$ are represented in Figures 4.10(d)–(f), where the additive noise is considered as white and Gaussian and for which the SNR = −10 dB. The classical Wiener filtering of parameter K (Wcc-K) of each component, with a signal subspace dimension fixed to $K = 8$, permits us to obtain the results presented in Figures 4.10(g)–(i). The multimode PCA-based filtering achieved by applying HOSVD-(8,8,3) to noisy data tensor permits to obtain the results presented in Figures 4.10(j)–(l). Finally, the results obtained with multiway Wiener filtering applied to the noisy data tensor are presented in Figures 4.10(m)–(o). The evolution of the NQE with respect to the SNR

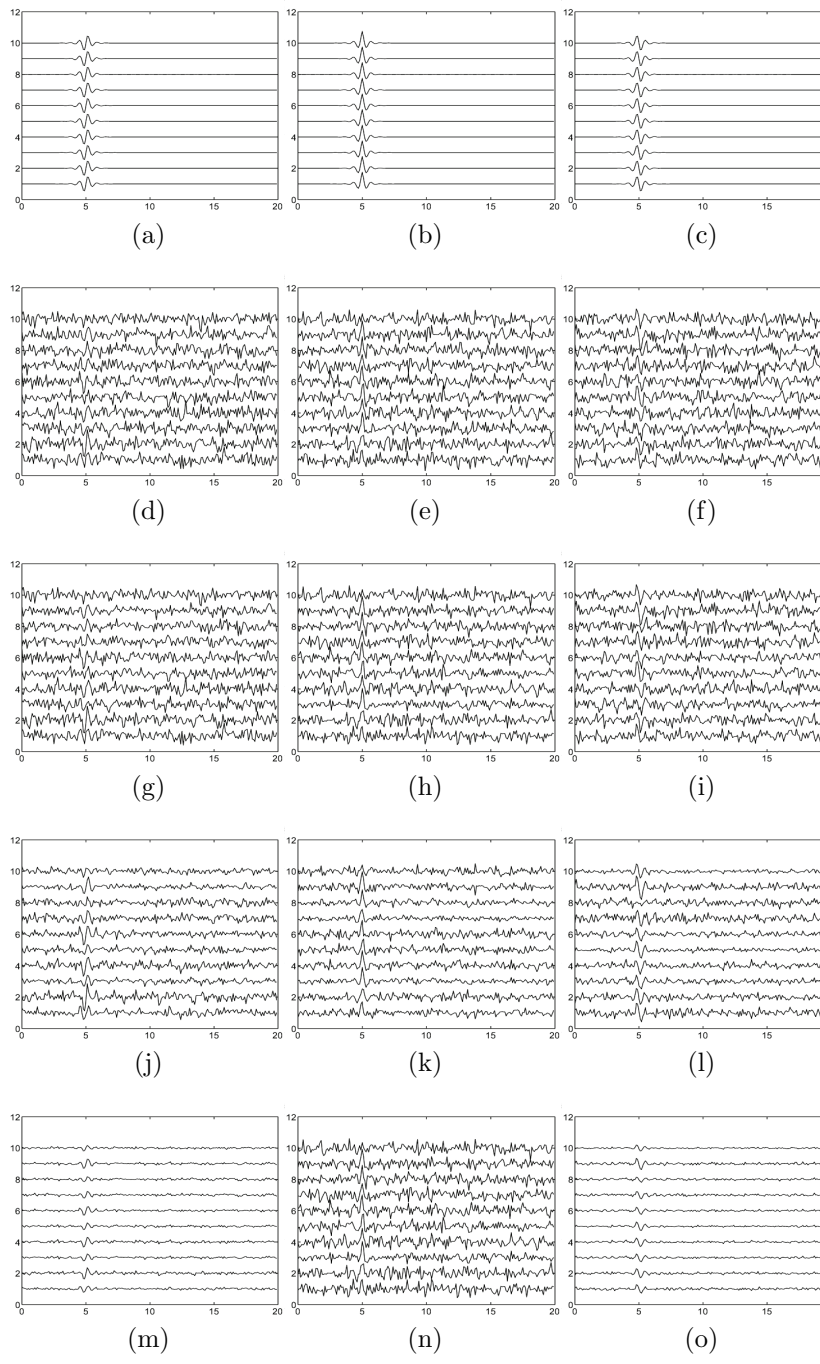
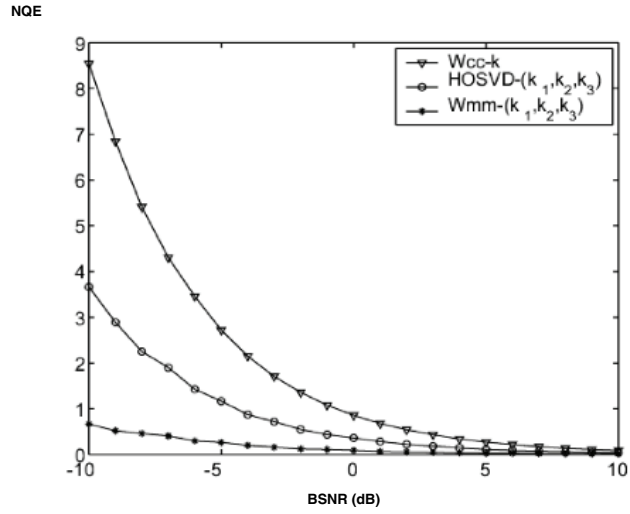
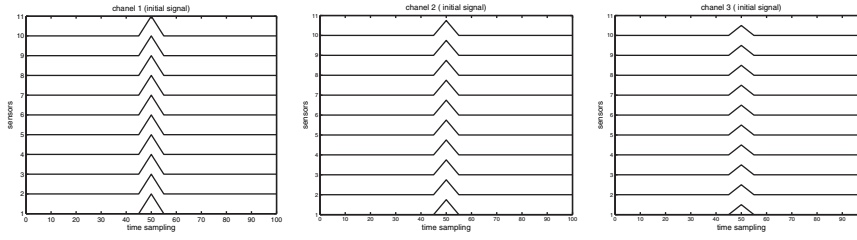


FIG. 4.10. *Nonnoisy, impaired, and processed seismic wave: the three polarization components.* (a)–(c) *Components* $1, 2,$ *and* 3 *of the nonnoisy seismic wave.* (d)–(f) *Components* $1, 2,$ *and* 3 *of the seismic wave, impaired by an additive white Gaussian noise (SNR* $= -10$ *dB).* (g)–(i) *Wiener filtering applied component by component (Wcc-K), with rank* $K = 8$. (j)–(l) *HOSVD-(*K_1, K_2, K_3*), with* $(K_1, K_2, K_3) = (8, 8, 3)$. (m)–(o) *multiway Wiener filtering (Wmm-(*K_1, K_2, K_3*)), with* $(K_1, K_2, K_3) = (8, 8, 3)$.



(a)

FIG. 4.11. *Evolution of the NQE with respect to the SNR (dB) for each tensor filtering method. (∇): Wiener filtering applied component by component (Wcc-K), with rank* $K = 8$*; (∘): HOSVD-*(K_1, K_2, K_3)*, with* $(K_1, K_2, K_3) = (8, 8, 3)$*; (•): multiway Wiener filtering (Wmm-*(K_1, K_2, K_3)*) with* $(K_1, K_2, K_3) = (8, 8, 3)$.



(a) (b) (c)

FIG. 4.12. *Multicomponent seismic signal.* (a)–(c) *Components* 1 *to* 34 *of the nonnoisy seismic wave.*

(dB) is given in Figure 4.11. As well as in the case of color image filtering, in this simulation, the best quality, in terms of noise reduction, is given by multiway Wiener filtering since, for all considered SNR values, the NQE values given by this method are lower than the values given by both HOSVD-$(8, 8, 3)$ and Wcc-8.

4.5.2. Filtering of a multicomponent seismic type signal impaired by an additive correlated Gaussian noise. In this simulation, we consider a multicomponent seismic wave, impaired by a correlated Gaussian noise. The purpose here is to compare the perfomances of multiway filtering algorithms based on either second order moments or fourth order cumulants. Figures 4.12 and 4.13 show the efficiency, in terms of noise reduction, of rank-$\mathcal{C}(K_1, K_2, K_3)$ based filtering and rank-$\mathbf{C_1}(K_1, K_2, K_3)$ based filtering compared to rank-(K_1, K_2, K_3) approximation based on second order statistics, when seismic signals impaired by a correlated Gaussian noise are considered.

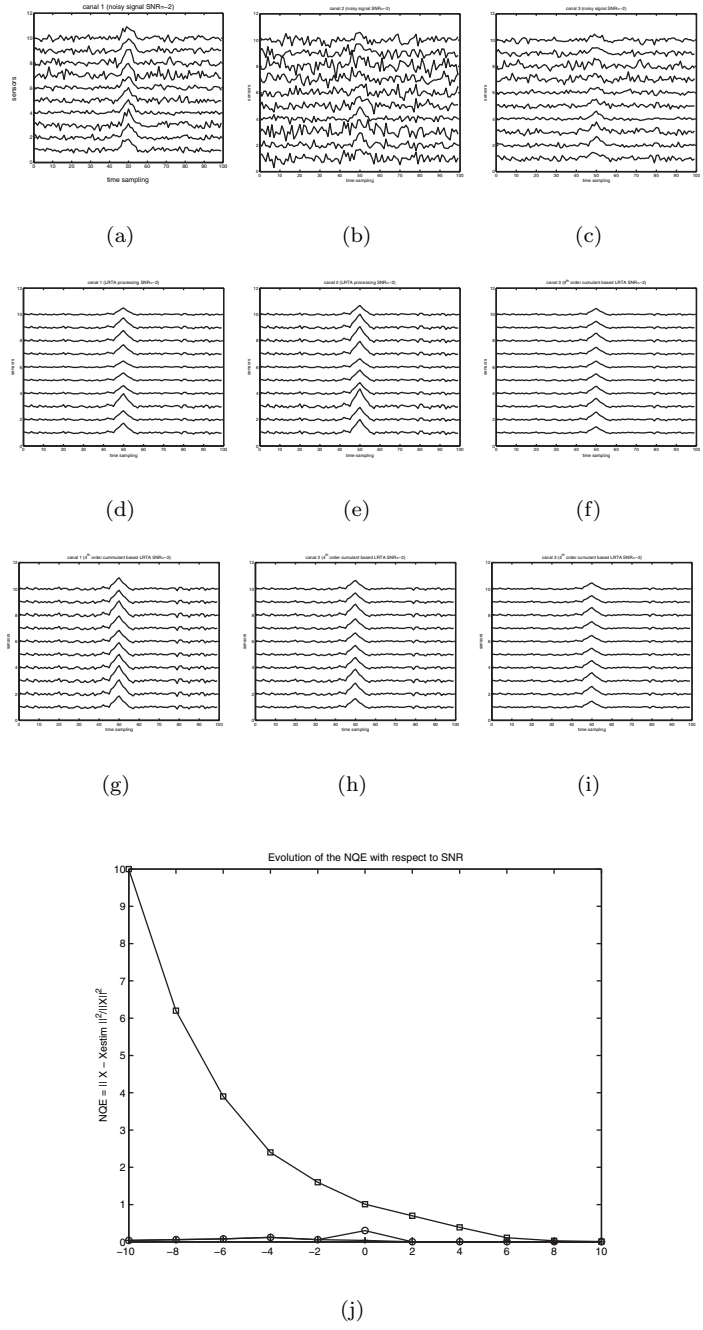


FIG. 4.13. *Denoising of a multicomponent seismic wave impaired by an additive correlated Gaussian noise (SNR= −2 dB), using multiway filtering based on fourth order cumulants: comparison of rank–$\mathcal{C}(8,8,3)$, and rank–$\mathbf{C}_1(8,8,3)$.* (a)–(c) *Noised signal; components* 1 *to* 3 *impaired by a correlated Gaussian noise (SNR= −2 dB).* (d)–(f) *rank-$\mathcal{C}(8,8,3)$ based filtering.* (g)–(i) *rank-$\mathbf{C}_1(8,8,3)$ based filtering.* (j) *Evolution of NQE with respect to SNR (dB) for rank-$(8,8,3)$ approximation (□), rank-$\mathbf{C}_1(8,8,3)$ using fourth order cumulant slice matrix (∘), and rank-$\mathcal{C}(8,8,3)$ using fourth order cumulants (+).*

5. Conclusion. In this paper, an overview on new mathematical methods dedicated to multicomponent data is presented. Multicomponent data are represented as tensors, that is, multiway arrays, and the tensor filtering methods that are presented rely on multilinear algebra. First we present how to perform channel-by-channel SVD-based filtering. Then we review three methods that take into account the relationships between each component of a processed tensor. The first method consists of an extension of the classical SVD-based filtering method. In the case of an additive white Gaussian noise, the signal tensor is estimated thanks to a multimode PCA achieved by applying a lower rank-$(K_1, \ldots, K_N)$ approximation to the noisy data tensor, or a lower rank-$(K_1, \ldots, K_N)$ truncation of its HOSVD. This method is implicitly based on second order statistics and relies on the orthogonality between nth-mode noise and signal subspaces. The second presented method consists of an improvement of the multimode PCA-based tensor filtering in the case of an additive correlated Gaussian noise. In this case, the covariance matrix involved in TUCKALS3 algorithm is replaced with the fourth order cumulant matrix of the related vectors. We reviewed a low computational load procedure involving the fourth order cumulant slice matrix instead of fourth order cumulants. This improved multimode PCA provides good performances compared to the multimode PCA method based on second order statistics, as was shown in the case of noise reduction in color images and multicomponent seismic waves.

Finally, the third reviewed method is a multiway version of the classical Wiener filtering. In extension to the one-dimensional case, the nth-mode Wiener filters are estimated by minimizing the mean squared error between the expected signal tensor and the estimated signal tensor obtained by applying the nth-mode Wiener filters to the noisy data tensor thanks to the nth-mode product operator. An alternating least squares algorithm has been presented to determine the optimal nth-mode Wiener filters. The performances of this multiway Wiener filtering and comparative results with multimode PCA have been presented in the case of additive white noise reduction in a color image and in a multicomponent seismic wave.

Appendix A. nth-mode Wiener filter analytical expression. The following computations are related to section 3.3. They rely on the definitions and properties of tensors and multilinear algebra that can be found in [11, 13, 14].

The mean squared error involved in multiway Wiener filtering is given by relation

$$\begin{aligned} \mathrm{e}(\mathbf{H}^{(1)}, \ldots, \mathbf{H}^{(N)}) = {} & \mathrm{E}\left[\|\mathcal{X}\|^2\right] - 2\mathrm{E}\left[\left\langle \mathcal{X}, \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \cdots \times_N \mathbf{H}^{(N)} \right\rangle\right] \\ & + \mathrm{E}\left[\left\| \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \cdots \times_N \mathbf{H}^{(N)} \right\|^2\right]. \end{aligned} \tag{A.1}$$

The Frobenius norm of a tensor is also equal to the norm of any of its nth-mode flattening matrices. In order to determine the expression of filter $\mathbf{H}^{(n)}$ associated with fixed filters $\mathbf{H}^{(m)}$, for all $m \neq n$, the nth-mode flattening of (A.1) is processed.

Let us define matrix $\mathbf{F}^{(n)}_{\mathbf{XR}}$ as

$$\mathbf{F}^{(n)}_{\mathbf{XR}} = \mathbf{X}_n \mathbf{T}^{(n)} \mathbf{R}_n^T \tag{A.2}$$

with

$$\mathbf{T}^{(n)} = \mathbf{H}^{(1)} \otimes \cdots \otimes \mathbf{H}^{(n-1)} \otimes \mathbf{H}^{(n+1)} \otimes \cdots \otimes \mathbf{H}^{(N)}. \tag{A.3}$$

Hence, for all $n = 1$ to N,

$$\left\langle \mathcal{X}, \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \cdots \times_N \mathbf{H}^{(N)} \right\rangle = \mathrm{tr}\left(\mathbf{F}^{(n)}_{\mathbf{XR}} {\mathbf{H}^{(n)}}^T \right). \tag{A.4}$$

Let us define matrix $\mathbf{G}_{\mathbf{RR}}^{(n)}$ as

$$\mathbf{G}_{\mathbf{RR}}^{(n)} = \mathbf{R}_n \mathbf{Q}^{(n)} {\mathbf{R}_n}^T \tag{A.5}$$

with

$$\mathbf{Q}^{(n)} = \mathbf{T}^{(n)T} \mathbf{T}^{(n)}$$

$$\mathbf{Q}^{(n)} = {\mathbf{H}^{(1)}}^T \mathbf{H}^{(1)} \otimes \cdots \otimes {\mathbf{H}^{(n-1)}}^T \mathbf{H}^{(n-1)} \otimes {\mathbf{H}^{(n+1)}}^T \mathbf{H}^{(n+1)} \otimes \cdots \otimes {\mathbf{H}^{(N)}}^T \mathbf{H}^{(N)}. \tag{A.6}$$

Hence for all $n = 1$ to N,

$$\left\| \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \cdots \times_N \mathbf{H}^{(N)} \right\|^2 = \mathrm{tr}\left(\mathbf{H}^{(n)} \mathbf{G}_{\mathbf{RR}}^{(n)} {\mathbf{H}^{(n)}}^T \right). \tag{A.7}$$

Minimization of mean squared error $\mathbf{e(H^{(1)}, \ldots, H^{(N)})}$. The expression of the nth-mode flattened mean squared error $\mathrm{e}(\mathbf{H}^{(1)}, \ldots, \mathbf{H}^{(N)})$ is the following:

$$\begin{aligned} \mathrm{e}(\mathbf{H}^{(1)}, \ldots, \mathbf{H}^{(N)}) = \mathrm{E}\left[\|\mathbf{X}_n\|^2 \right] - 2\mathrm{E}\left[\mathrm{tr}\left(\mathbf{F}_{\mathbf{XR}}^{(n)} {\mathbf{H}^{(n)}}^T \right) \right] \\ + \mathrm{E}\left[\mathrm{tr}\left(\mathbf{H}^{(n)} \mathbf{G}_{\mathbf{RR}}^{(n)} {\mathbf{H}^{(n)}}^T \right) \right]. \end{aligned} \tag{A.8}$$

Assuming that m-mode filters $\mathbf{H}^{(m)}$ are fixed for all $m \neq n$, mean squared error $\mathrm{e}(\mathbf{H}^{(1)}, \ldots, \mathbf{H}^{(N)})$ is minimal when its gradient with respect to nth-mode filter $\mathbf{H}^{(n)}$ is null,

$$\mathbf{grad}(\mathrm{e}) = \left[\frac{\partial \mathrm{e}}{\partial \mathbf{H}^{(1)}}, \ldots, \frac{\partial \mathrm{e}}{\partial \mathbf{H}^{(N)}} \right]^T, \tag{A.9}$$

that is, when $\frac{\partial \mathrm{e}}{\partial \mathbf{H}^{(n)}}$ are conjointly null for all $n = 1$ to N. Let us study $\frac{\partial \mathrm{e}}{\partial \mathbf{H}^{(n)}}$ for a given nth-mode. The nth-mode filters $\mathbf{H}^{(m)}$ are supposed to be fixed for all $m \in \{1, \ldots, N\} - \{n\}$. Then $\frac{\partial \mathrm{e}}{\partial \mathbf{H}^{(n)}} = 0$ implies that

$$\mathrm{E}\left[\frac{\partial}{\partial \mathbf{H}^{(n)}} \mathrm{tr}\left(\mathbf{H}^{(n)} \mathbf{G}_{\mathbf{RR}}^{(n)} {\mathbf{H}^{(n)}}^T \right) \right] = 2\mathrm{E}\left[\frac{\partial}{\partial \mathbf{H}^{(n)}} \mathrm{tr}\left(\mathbf{F}_{\mathbf{XR}}^{(n)} {\mathbf{H}^{(n)}}^T \right) \right], \tag{A.10}$$

We compute then the derivatives on both sides in (A.10), taking into account the fact that $\mathbf{G}_{\mathbf{RR}}^{(n)}$ and $\mathbf{F}_{\mathbf{XR}}^{(n)}$ are independent from $\mathbf{H}^{(n)}$:

$$\frac{\partial}{\partial \mathbf{H}^{(n)}} \mathrm{tr}\left(\mathbf{F}_{\mathbf{XR}}^{(n)} {\mathbf{H}^{(n)}}^T \right) = \mathbf{F}_{\mathbf{XR}}^{(n)}, \tag{A.11}$$

$$\frac{\partial}{\partial \mathbf{H}^{(n)}} \mathrm{tr}\left(\mathbf{H}^{(n)} \mathbf{G}_{\mathbf{RR}}^{(n)} {\mathbf{H}^{(n)}}^T \right) = 2\mathbf{H}^{(n)} \mathbf{G}_{\mathbf{RR}}^{(n)}. \tag{A.12}$$

Expression of $\mathbf{H}^{(\ )}$, nth-mode Wiener filter. Replacing (A.11) and (A.12) into expression (A.10) leads to the expression of $\mathbf{H}^{(n)}$ nth-mode Wiener filter associated with fixed $\mathbf{H}^{(m)}$ m-mode filters, $m \neq n$:

$$\mathbf{H}^{(n)} = \gamma_{\mathbf{XR}}^{(n)} {\mathbf{\Gamma}_{\mathbf{RR}}^{(n)}}^{-1}, \tag{A.13}$$

where

$$\gamma_{\mathbf{XR}}^{(n)} = \mathrm{E}\left[\mathbf{F}_{\mathbf{XR}}^{(n)} \right] \tag{A.14}$$

is the $\mathbf{T}^{(n)}$-weighted covariance matrix between the signal $\mathbf{X}_n$ and the data $\mathbf{R}_n$ and

$$\mathbf{\Gamma}^{(n)}_{\mathbf{RR}} = \mathrm{E}\left[\mathbf{G}^{(n)}_{\mathbf{RR}}\right] \tag{A.15}$$

is the $\mathbf{Q}^{(n)}$-weighted correlation matrix of the data.

Appendix B. Assumptions and related expression of the *n*th-mode Wiener filter. The following computations are related to section 3.3. Let us consider matrices $\mathbf{T}^{(n)}$ and $\mathbf{Q}^{(n)}$ defined in (A.3) and (A.6). Their generic (i,j)-terms are denoted respectively by $T^{(n)}_{ij}$ and by $Q^{(n)}_{ij}$.

Weight matrix term independence. The terms of weight matrix $\mathbf{O}^{(n)} \in \mathbb{R}^{K_n \times M_n}$ are supposed mutually independent,

$$\mathrm{E}\left[o_{kl}o_{mn}\right] = \alpha_{kl}\delta_{km}\delta_{ln}, \tag{B.1}$$

whatever k and $m \in \{1,\ldots,K_n\}$, l and $n \in \{1,\ldots,M_n\}$ and where α_{kl} is not null.

White and Gaussian noise condition. White and Gaussian noise condition applied to the nth-mode flattening $\mathbf{N}_n$ can be expressed by

$$\mathrm{E}\left[n_{kl}n_{pq}\right] = \sigma_n^2\delta_{kp}\delta_{lq}, \tag{B.2}$$

where $(k,p) \in \{1,\ldots,K_n\}^2$, $(l,q) \in \{1,\ldots,M_n\}^2$ and σ_n^2 is the nth-mode noise power.

Noise and signal independence. The condition on noise and signal independence can be expressed by

$$\mathrm{E}\left[x_{kl}n_{pq}\right] = 0 \tag{B.3}$$

for all $(k,p) \in \{1,\ldots,K_n\}^2$ and $(l,q) \in \{1,\ldots,M_n\}^2$. Hence, $\mathbf{T}^{(n)}$ and $\mathbf{Q}^{(n)}$-weighted $(\mathbf{X},\mathbf{N})$-covariance matrices are null:

$$\begin{aligned} \gamma^{(n)}_{\mathbf{XN}} &= \gamma^{(n)}_{\mathbf{NX}} = \mathbf{0}, \\ \mathbf{\Gamma}^{(n)}_{\mathbf{XN}} &= \mathbf{\Gamma}^{(n)}_{\mathbf{NX}} = \mathbf{0}. \end{aligned} \tag{B.4}$$

Indeed, their (i,j)-term is

$$\begin{aligned} \left(\gamma^{(n)}_{\mathbf{XN}}\right)_{ij} &= \sum_{k=1}^{M_n}\sum_{l=1}^{M_n} T^{(n)}_{kl}\mathrm{E}\left[x_{ik}n_{jl}\right], \\ \left(\mathbf{\Gamma}^{(n)}_{\mathbf{XN}}\right)_{ij} &= \sum_{k=1}^{M_n}\sum_{l=1}^{M_n} Q^{(n)}_{kl}\mathrm{E}\left[x_{ik}n_{jl}\right]. \end{aligned} \tag{B.5}$$

Expressions of weighted covariance matrices.

Covariance matrix $\gamma^{(\ )}_{\mathbf{RR}}$. As $\mathbf{R}_n = \mathbf{X}_n + \mathbf{N}_n$, the expression of $\gamma^{(n)}_{\mathbf{RR}}$ reads

$$\gamma^{(n)}_{\mathbf{RR}} = \gamma^{(n)}_{\mathbf{XX}} + \gamma^{(n)}_{\mathbf{XN}} + \gamma^{(n)}_{\mathbf{NX}} + \gamma^{(n)}_{\mathbf{NN}}. \tag{B.6}$$

So according to (B.4), $\gamma^{(n)}_{\mathbf{RR}}$ weighted covariance matrix can be expressed by

$$\gamma^{(n)}_{\mathbf{RR}} = \gamma^{(n)}_{\mathbf{XX}} + \gamma^{(n)}_{\mathbf{NN}}. \tag{B.7}$$

Moreover,

$$\gamma^{(n)}_{\mathbf{XR}} = \gamma^{(n)}_{\mathbf{XX}} + \gamma^{(n)}_{\mathbf{XN}} = \gamma^{(n)}_{\mathbf{XX}}. \tag{B.8}$$

Covariance matrix $\mathbf{\Gamma}^{(\,)}_{\mathbf{RR}}$. Relations (B.6), (B.7), and (B.8) hold as well for $\mathbf{\Gamma}^{(n)}_{\mathbf{RR}}$:

$$\mathbf{\Gamma}^{(n)}_{\mathbf{RR}} = \mathbf{\Gamma}^{(n)}_{\mathbf{XX}} + \mathbf{\Gamma}^{(n)}_{\mathbf{XN}} + \mathbf{\Gamma}^{(n)}_{\mathbf{NX}} + \mathbf{\Gamma}^{(n)}_{\mathbf{NN}}$$

and

$$\mathbf{\Gamma}^{(n)}_{\mathbf{RR}} = \mathbf{\Gamma}^{(n)}_{\mathbf{XX}} + \mathbf{\Gamma}^{(n)}_{\mathbf{NN}}. \tag{B.9}$$

Moreover,

$$\mathbf{\Gamma}^{(n)}_{\mathbf{XR}} = \mathbf{\Gamma}^{(n)}_{\mathbf{XX}} + \mathbf{\Gamma}^{(n)}_{\mathbf{XN}} = \mathbf{\Gamma}^{(n)}_{\mathbf{XX}}. \tag{B.10}$$

Expressions of $\mathbf{\Gamma}^{(\,)}_{\mathbf{NN}}$ and $\boldsymbol{\gamma}^{(\,)}_{\mathbf{NN}}$. According to (B.2), the (i,j)-term of $\mathbf{\Gamma}^{(n)}_{\mathbf{NN}}$ is the following:

$$\left(\mathbf{\Gamma}^{(n)}_{\mathbf{NN}}\right)_{ij} = \sum_{k=1}^{M_n}\sum_{l=1}^{M_n} Q^{(n)}_{kl}\mathrm{E}[n_{ik}n_{jl}] = {\sigma^{(n)}_{\Gamma}}^2 \delta ij \tag{B.11}$$

with

$${\sigma^{(n)}_{\Gamma}}^2 = \mathrm{tr}(\mathbf{Q}^{(n)})\sigma_n^2. \tag{B.12}$$

Hence

$$\mathbf{\Gamma}^{(n)}_{\mathbf{NN}} = {\sigma^{(n)}_{\Gamma}}^2 \mathbf{I}_{I_n}. \tag{B.13}$$

The (i,j)-term of $\gamma^{(n)}_{\mathbf{NN}}$ can also be expressed by

$$\left(\gamma^{(n)}_{\mathbf{NN}}\right)_{ij} = \sum_{k=1}^{M_n}\sum_{l=1}^{M_n} T^{(n)}_{kl}\mathrm{E}[n_{ik}n_{jl}] = {\sigma^{(n)}_{\gamma}}^2 \delta ij$$

with

$${\sigma^{(n)}_{\gamma}}^2 = \mathrm{tr}(\mathbf{T}^{(n)})\sigma_n^2.$$

Hence

$$\gamma^{(n)}_{\mathbf{NN}} = {\sigma^{(n)}_{\gamma}}^2 \mathbf{I}_{I_n}. \tag{B.14}$$

Expressions of $\mathbf{\Gamma}^{(\,)}_{\mathbf{XX}}$ and $\boldsymbol{\gamma}^{(\,)}_{\mathbf{XX}}$. Considering the signal model (3.16),

$$\gamma^{(n)}_{\mathbf{XX}} = \mathbf{V}^{(n)}_s \gamma^{(n)}_{\mathbf{OO}} {\mathbf{V}^{(n)}_s}^T, \tag{B.15}$$

where

$$\gamma^{(n)}_{\mathbf{OO}} = \mathrm{E}\left[\mathbf{O}^{(n)}\mathbf{T}^{(n)}{\mathbf{O}^{(n)}}^T\right]. \tag{B.16}$$

According to (B.1), the generic term of $\gamma^{(n)}_{\mathbf{OO}}$ is

$$\left(\gamma^{(n)}_{\mathbf{OO}}\right)_{ij} = \sum_{k=1}^{M_n}\sum_{l=1}^{M_n} T^{(n)}_{kl}\mathrm{E}\left[n_{ik}n_{jl}\right] = \beta_i\delta_{ij}, \tag{B.17}$$

where, for all $i = 1$ to K_n,

$$\beta_i = \sum_{k=1}^{M_n} T_{kk}^{(n)} \alpha_{ik}, \tag{B.18}$$

and where α_{ik} is defined in (B.1). So, $\gamma_{\mathbf{OO}}^{(n)}$ is a diagonal matrix:

$$\gamma_{\mathbf{OO}}^{(n)} = \begin{bmatrix} \beta_1 & & 0 \\ & \ddots & \\ 0 & & \beta_{K_n} \end{bmatrix}. \tag{B.19}$$

The matrix $\mathbf{\Gamma}_{\mathbf{XX}}^{(n)}$ is also expressed as

$$\mathbf{\Gamma}_{\mathbf{XX}}^{(n)} = \mathbf{V}_s^{(n)} \mathbf{\Gamma}_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)^T}, \tag{B.20}$$

where $\mathbf{\Gamma}_{\mathbf{OO}}^{(n)}$ is the diagonal matrix

$$\mathbf{\Gamma}_{\mathbf{OO}}^{(n)} = \begin{bmatrix} \epsilon_1 & & 0 \\ & \ddots & \\ 0 & & \epsilon_{K_n} \end{bmatrix}, \tag{B.21}$$

and

$$\epsilon_i = \sum_{k=1}^{M_n} Q_{kk}^{(n)} \alpha_{ik}, \tag{B.22}$$

where α_{ik} is defined in (B.1).

Final expression of $\mathbf{H}^{(\ )}$, nth-mode Wiener filter. According to (B.8) and (B.15),

$$\gamma_{\mathbf{XR}}^{(n)} = \mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)^T}. \tag{B.23}$$

According to (B.9), (B.13), and (B.20),

$$\mathbf{\Gamma}_{\mathbf{RR}}^{(n)} = \mathbf{V}_s^{(n)} \mathbf{\Gamma}_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)^T} + {\sigma_\Gamma^{(n)}}^2 \mathbf{I}_{I_n},$$

which can be expressed as

$$\mathbf{\Gamma}_{\mathbf{RR}}^{(n)} = \begin{bmatrix} \mathbf{V}_s^{(n)} \mathbf{V}_b^{(n)} \end{bmatrix} \begin{bmatrix} \mathbf{\Gamma}_{\mathbf{OO}}^{(n)} + {\sigma_\Gamma^{(n)}}^2 \mathbf{I}_{K_n} & 0 \\ 0 & {\sigma_\Gamma^{(n)}}^2 \mathbf{I}_{I_n - K_n} \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^{(n)^T} \\ \mathbf{V}_b^{(n)^T} \end{bmatrix} \tag{B.24}$$

with $\mathbf{V}_b^{(n)} \in \mathrm{St}(I_n, I_n - K_n)$ the columnwise orthogonal matrix containing the noise subspace basis vectors. The assumption of noise and signal independence implies that the noise and signal subspaces are orthogonal:

$$\mathbf{V}_s^{(n)^T} \mathbf{V}_b^{(n)} = \mathbf{0}. \tag{B.25}$$

Let us call

$$\mathbf{\Lambda}_s^{(n)} = \mathbf{\Gamma}_{\mathbf{OO}}^{(n)} + {\sigma_\Gamma^{(n)}}^2 \mathbf{I}_{K_n} \tag{B.26}$$

and

$$\mathbf{\Lambda}_b^{(n)} = {\sigma_{\mathbf{\Gamma}}^{(n)}}^2 \mathbf{I}_{I_n - K_n}. \tag{B.27}$$

Inserting the last expressions of $\gamma_{\mathbf{XR}}^{(n)}$ and $\mathbf{\Gamma}_{\mathbf{RR}}^{(n)}$ (see (B.23) and (B.24)) into Wiener nth-mode filter expression (A.13) leads to

$$\mathbf{H}^{(n)} = \mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)^T} \left[\mathbf{V}_s^{(n)} \, \mathbf{V}_b^{(n)} \right] \begin{bmatrix} {\Lambda_s^{(n)}}^{-1} & 0 \\ 0 & {\Lambda_b^{(n)}}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^{(n)^T} \\ \mathbf{V}_b^{(n)^T} \end{bmatrix}, \tag{B.28}$$

which can be expressed as

$$\mathbf{H}^{(n)} = \left[(\mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)^T} \mathbf{V}_s^{(n)}) \; (\mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)^T} \mathbf{V}_b^{(n)}) \right] \begin{bmatrix} {\mathbf{\Lambda}_s^{(n)}}^{-1} \mathbf{V}_s^{(n)^T} & 0 \\ 0 & {\mathbf{\Lambda}_b^{(n)}}^{-1} \mathbf{V}_b^{(n)^T} \end{bmatrix} \tag{B.29}$$

Considering noise and signal orthogonality condition (B.25) and the fact that $\mathbf{V}_n^{(n)} \mathbf{V}_n^{(n)^T} = \mathbf{I}_{K_n}$, the final Wiener nth-mode filter expression becomes

$$\mathbf{H}^{(n)} = \mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} {\mathbf{\Lambda}_{\mathbf{\Gamma s}}^{(n)}}^{-1} \mathbf{V}_s^{(n)^T}. \tag{B.30}$$

Acknowledgments. We would like to thank the anonymous reviewers who contributed to the quality of this paper by providing helpful suggestions.

REFERENCES

[1] K. Abed-Meraim, H. Maître, and P. Duhamel, *Blind multichannel image restoration using subspace based method*, in Proceedings of the IEEE International Conference on Accoustics Systems and Signal Processing, Hong Kong, China, 2003.

[2] O. Alter and G. H. Golub, *Reconstructing the pathways of a cellular system from genome-scale signals by using matrix and tensor computations*, Proc. Natl. Acad. Sci. USA, 102 (2005), pp. 17559–17564.

[3] H. C. Andrews and C. L. Patterson, *Singular value decomposition and digital image processing*, IEEE Trans. Acoustics, Speech, and Signal Processing, 24 (1976), pp. 26–53.

[4] H. C. Andrews and C. L. Patterson, *Singular Value Decomposition (SVD) image coding*, IEEE Trans. Communications, 1976, pp. 425–432.

[5] B. W. Bader and T. G. Kolda, *Algorithm* 862: *Matlab tensor classes for fast algorithm prototyping*, ACM Tran. Math. Software, 32 (2006).

[6] E. Bala and A. Ertüzün, *A multivariate thresholding technique for image denoising using multiwavelets*, EURASIP J. ASP, 8 (2005), pp. 1205–1211.

[7] S. Bourennane and A. Bendjama, *Locating wide band acoustic sources using higher order statistics*, Appl. Acoustics, 63 (2001), pp. 235–251.

[8] A. Benjama, S. Bourennane, and M. Frikel, *Seismic wave separation based on higher order statistics*, in Proceedings of the PIEEE International Conference on Digital Signal Processing and its Applications, Moscow, Russia, 1998.

[9] R. Bro, *Multi-way Analysis in the Food Industry*, Ph.D. thesis, Royal Veterinary and Agricultural University, Denmark, 1998.

[10] J. D. Caroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart–Young decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[11] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*, Ph.D. thesis, K. U. Leuven, E. E. Department (ESAT), Belgium, 1997.

[12] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Dimensionality reduction in higher-order-only ICA*, Proc. IEEE Signal Processing Workshop on Higher Order Statistics, 7 (1997), pp. 316–320.

[13] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[14] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-(1) and rank-$(r_1, \ldots, r_N)$ approximation of higher-order tensors,* SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.
[15] Y. Eckart and G. Young, *The approximation of a matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.
[16] F. L. M. Freire and T. J. Ulrych, *Application of SVD to vertical seismic profiling*, Geophysics, 53 (1988, pp. 778–785.
[17] F. Glangeaud and J.-L. Mari, *Wave Separation*, Technip IFP, 1994.
[18] G. H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., John Hopkins University Press, Baltimore, 1996.
[19] E. Gonen and J. M. Mendel, *Application of cumulants to array processing Part* III*: Blind beamforming for coherent signal*, IEEE Trans. Signal Process., 45 (1997), pp. 2252–64.
[20] R. A. Harshman and M. E. Lundy, *Research methods for multimode data analysis*, Praeger, New York, 1984, pp. 122–215.
[21] M. Hemon and D. Mace, *The use of Karhunen–Loeve transform in seismic data prospecting,* Geophys. Prospecting, 26 (1978), pp. 600–626.
[22] J. Håstad, *Tensor rank is np-complete*, J. Algorithms, 11 (1990), pp. 644–654.
[23] K. Hsu, *Wave separation and feature extraction of accoustic well-logging waveforme of triaxial recordings by singular value decomposition*, Geophysics, 55 (1990), pp. 176–184.
[24] G. M. Jackson, I. M. Mason, and S. A. Greenhalgh, *Principal component transforms of triaxial recordings by singular value decomposition*, Geophysics, 56 (1991), pp. 176–184.
[25] H. Kiers, *Towards a standardized notation and terminology in multiway analysis*, J. Chemometrics, 14 (2000), pp. 105–122.
[26] E. Kofidis and P. A. Regalia, *On the best rank-*1 *approximation of higher-order supersymetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 863–884.
[27] T. G. Kolda, *Orthogonal tensor decomposition*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 243–255.
[28] P. M. Kroonenberg, *Three-mode Principal Component Analysis*, DSWO Press, Leiden, Netherlands, 1983.
[29] P. M. Kroonenberg and J. De Leeuw, *Principal component analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 45 (1980), pp. 69–97.
[30] J. B. Kruskal, *Rank, decomposition, and uniqueness for* 3*-way and* N*-way arrays*, Multiway Data Analysis, Elsevier, Amsterdam, Netherlands, 1988.
[31] J. L. Lacoume, P. O. Amblard, and P. Comon, *Statistiques d'ordre supérieur pour le traitement du signal*, Masson, Paris, 1997.
[32] X. Liu, *Ground roll suppression using the Karhunen–Loeve transform*, Geophysics, 64 (1991), pp. 564–566.
[33] J. M. Mendel, *Tutorial on higher order statistics (spectra) in signal processing and system theory: Theoretical results and some applications*, Proc. IEEE, 79 (1991), pp. 278–305.
[34] D. Muti, *Traitement du Signal Tensoriel. Application aux images en Couleurs et aux Signaux Sismiques,* Ph.D. thesis, Institut Fresnel, Université Paul Cézanne, Aix-Marseille III, Marseille, France, 2004.
[35] D. Muti and S. Bourennane, *Multidimensional estimation based on a tensor decomposition*, in Proceedings of the IEEE Workshop on Statistical Signal Processing, St. Louis, 2003.
[36] D. Muti and S. Bourennane, *Multidimensional signal processing using lower rank tensor approximation*, in Proceedings of the IEEE International Conference on Accoustics, Systems, and Signal Processing, Hong Kong, China, 2003.
[37] D. Muti and S. Bourennane, *Traitement du signal par décomposition tensorielle*, in Proceedings of the GRETSI Symposium, Paris, France, 2003.
[38] D. Muti and S. Bourennane, *Multidimensional filtering based on a tensor approach*, Signal Process. J., 85 (2005), pp. 2338–2353.
[39] D. Muti and S. Bourennane, *Multiway filtering based on fourth order cumulants*, Appl. Signal Process., 7 (2005), pp. 1147–1159.
[40] R. Neelamani, H. Choi, and R. Baraniuk, *Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems*, IEEE Trans. Signal Process., 52 (2004), pp. 418–433.
[41] B. Porat and B. Friedlander, *Direction finding algorithms based on higher-order statistics*, IEEE Trans. Signal Process., 39 (1991), pp. 2016–2024.
[42] C. Preza, M. I. Miller, L. J. Thomas Jr., and J. G. McNally, *Regularized method for reconstruction of three-dimensional microscopic objects from optical sections*, J. Opt. Soc. Amer. A, 9 (1992), pp. 219–228.
[43] L. W. She and B. Zheng, *Multiwavelets based denoising of SAR images*, in Proceedings of the 5th International Conference on Signal Processing, Beijing, China, 8, 2000, pp. 321–324.

[44] N. D. Sidiropoulos and R. Bro, *On the uniqueness of multilinear decomposition of N-way arrays*, J. Chemometrics, 14 (2000), pp. 229–239.
[45] N. D. Sidiropoulos, R. Bro, and G. Giannakis, *Parallel factor analysis in sensor array processing*, IEEE Trans. Signal Process., 48 (2000), pp. 2377–2388.
[46] N. D. Sidiropoulos, G. Giannakis, and R. Bro, *Blind PARAFAC receivers for DS-CDMA systems*, IEEE Trans. Signal Process., 48 (2000), pp. 810–823.
[47] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis: Applications in the Chemical Sciences*, John Wiley, New York, 2004.
[48] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
[49] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear image analysis for facial recognition*, in Proceedings of the IEEE International Conference on Pattern Recognition (ICPR2002), Vol. 2, Quebec City, Canada, 2002.
[50] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear independent components analysis*, in Proceedings of Learning 2004, Snowbird, UT, 2004.
[51] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear independent components analysis*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05), 1 (2005), pp. 547–553.
[52] H. Wang and N. Ahuja, *Facial expression decomposition*, in Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV2003), Vol. 2, Nice, France, 2003.
[53] M. Wax and T. Kailath, *Detection of signals information theoretic criteria*, IEEE Trans. Acoustics Speech Signal Process., 33, 1985.
[54] N. Yuen and B. Friedlander, *DOA estimation in multipath: An approach using fourth order cumulant*, IEEE Trans. Signal Process., 45 (1997), pp. 1253–63.
[55] N. Yuen and B. Friedlander, *Asymptotic performance analysis of blind signal copy using fourth order cumulant*, Internat. Adaptative Control Signal Process., 48 (1996), pp. 239–65.
[56] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensor*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1205–1218

NEWTON METHOD FOR JOINT APPROXIMATE DIAGONALIZATION OF POSITIVE DEFINITE HERMITIAN MATRICES*

MARCEL JOHO†

Abstract. In this paper we present a Newton method to jointly approximately diagonalize a set of positive definite Hermitian matrices. To this end, we derive the local gradient and Hessian of the underlying cost function in closed form. The algorithm is derived for the complex case and can also update a nonsquare diagonalization matrix. We analyze the cost function at the critical points and show its relation to a different cost function that is commonly studied.

Key words. joint approximate diagonalization, independent component analysis, Newton method, gradient and Hessian of a complex matrix-valued cost function

AMS subject classifications. 15A23, 49M15, 49Q12, 65F30

DOI. 10.1137/060659880

1. Problem formulation and cost function.

The mathematical problem of jointly approximately diagonalizing a set of P covariance matrices, $\{\mathbf{R}_p\}_{p=1}^{P}$, is of interest in statistical methods such as common principal component analysis [7]. Recently the joint diagonalization problem has received more attention also in the community of independent component analysis and blind signal separation, as some problems that occur in those fields can be formulated as a joint diagonalization optimization problem. As an example, we can describe one class of blind signal separation problems as follows:

Let $\mathbf{s}(t)$ be an M-dimensional vector containing the time series of M mutually independent *source signals* that are mixed by a *mixing matrix* $\mathbf{A}^{N\times M}$, such that $\mathbf{x}(t) = \mathbf{A}\,\mathbf{s}(t)$ is an N-dimensional vector containing the time series of N *sensor signals*. In the blind signal separation problem the assumption is made that $\mathbf{s}(t)$ and $\mathbf{A}$ are unknown; only the signals at the sensors, $\mathbf{x}(t)$, are known. Furthermore, the assumption is made that the source signals are nonstationary and mutually independent. Hence, $\mathbf{R}_{\mathbf{ss}}(t) \triangleq E\{\mathbf{s}(t)\mathbf{s}^H(t)\}$ depends on t and has a diagonal structure. If we further assume that the source signals are nonstationary, then $\mathbf{R}_{\mathbf{ss}}(t)$ is time-varying but always has a diagonal structure. The correlation matrix of the sensor signals $\mathbf{R}_{\mathbf{xx}}(t) \triangleq E\{\mathbf{x}(t)\mathbf{x}^H(t)\} = \mathbf{A}\,E\{\mathbf{s}(t)\mathbf{s}^H(t)\}\mathbf{A}^H = \mathbf{A}\,\mathbf{R}_{\mathbf{ss}}(t)\mathbf{A}^H$ becomes also time-varying but has no diagonal structure in general. In order to recover the unknown source signals, we aim at finding a *separation matrix* $\mathbf{W}$ such that $\mathbf{u}(t) = \mathbf{W}\mathbf{x}(t)$ becomes an estimate of the original source signals $\mathbf{s}(t)$, aside from a possible scaling and permutation of the elements in $\mathbf{u}(t)$. In order for $\mathbf{u}(t)$ to become an estimate of $\mathbf{s}(t)$, the correlation matrix

$$\mathbf{R}_{\mathbf{uu}}(t) \triangleq E\{\mathbf{u}(t)\mathbf{u}^H(t)\} = \mathbf{W}\,\mathbf{R}_{\mathbf{xx}}(t)\mathbf{W}^H \tag{1}$$

has to be diagonal for all t. Assuming that $\mathbf{W}$ is time-invariant, $\mathbf{W}$ has to be of the following structure:

$$\mathbf{W} = \mathbf{D}\mathbf{P}\mathbf{A}^{-1} \tag{2}$$

*Received by the editors May 16, 2006; accepted for publication (in revised form) by P. Comon November 12, 2007; published electronically September 25, 2008.
http://www.siam.org/journals/simax/30-3/65988.html
†Bose Corp., Framingham, MA 01701 (joho@ieee.org).

or, if $\mathbf{A}^{N\times M}$ is a tall matrix ($N > M$),

$$\mathbf{W} = \mathbf{DPA}^{\#}, \tag{3}$$

where $\mathbf{D}$ is an arbitrary diagonal matrix, $\mathbf{P}$ is an arbitrary permutation matrix, and $\mathbf{A}^{\#}$ denotes the pseudoinverse of $\mathbf{A}$. If we take P snapshots of $\mathbf{R_{xx}}(t)$, $\mathbf{R}_p \triangleq \mathbf{R_{xx}}(t_p)$, where t_p is the time instance of the pth snapshot, then we can formulate the task of finding a proper $\mathbf{W}$ as the following *joint diagonalization problem*:

Given a set of P positive definite Hermitian matrices $\mathbf{R}_p$, find a single matrix $\mathbf{W}$ that approximately joint diagonalizes the whole set $\{\mathbf{R}_p\}$ such that $\mathbf{W}\mathbf{R}_p\mathbf{W}^H$ is diagonal for all p.

Perfect diagonalization is typically not possible for a set of random positive definite Hermitian matrices $\{\mathbf{R}_p\}$, unless $\{\mathbf{R}_p\}$ is a set of commuting matrices [9]. However, the set $\{\mathbf{W}\,\mathbf{R}_p\mathbf{W}^H\}$ can still be approximately jointly diagonalized subject to a given cost function $\mathcal{J}(\mathbf{W};\{\mathbf{R}_p\}_{p=1}^P)$ that measures the degree of joint diagonalization.

Several cost functions for the joint diagonalization problem have been published in the last decade. In [5] Cardoso and Souloumiac have proposed a joint diagonalization cost function and have given a very effective Jacobi-type algorithm that minimizes their cost function under the constraint that the diagonalization matrix $\mathbf{W}$ is unitary. The core idea of their algorithm has been used in JADE [4] and SOBI [1], both very popular blind signal separation algorithms.

In [17] Yeredor published an algorithm, AC-DC, that uses a different cost function and is based on a subspace-fitting formulation. One advantage is that no orthogonality constraints are imposed on the diagonalization matrix $\mathbf{W}$. However, $\mathbf{W}$ needs to be square and real. In [19] Yeredor, Ziehe, and Müller also derived an algorithm based on the *natural gradient* that works even with nonpositive definite matrices $\mathbf{R}_p$; however, $\mathbf{W}$ needs to be real and square. Yeredor has also described in [18] how to compute a good initial value $\mathbf{W}_0$ for any iterative algorithm.

Another joint diagonalization algorithm, FFDIAG, that seems to have a very fast convergence rate, was presented by Ziehe, Laskov, and Müller in [20]. FFDIAG also requires that $\mathbf{R}_p$ and $\mathbf{W}$ be real square matrices. An extension of FFDIAG where $\mathbf{W}$ can be nonsquare or complex has not been published so far.

Recently, Vollgraf and Obermayer [16] published an algorithm for the real case, QDIAG, that also works for a nonsquare $\mathbf{W}$. Their algorithm sequentially solves a quadratic subproblem, which avoids the appearance of any higher-order terms in their cost function. QDIAG seems to have a similar convergence performance as FFDIAG and is very appealing from a computational point of view for joint-diagonalizing large sets of matrices.

In [14], Pham presented an efficient joint diagonalization algorithm that imposes no optimization constraints on $\mathbf{W}$, except that $\mathbf{W}$ must be square ($N = M$). This algorithm is also a Jacobi-type algorithm. In contrast to most other joint diagonalization algorithms, which minimize a constraint optimization problem, Pham's algorithm is formulated to minimize an unconstrained optimization problem. The underlying cost function of this algorithm is based on some preliminary work by Flury [7] and Flury and Gautschi [8], namely,

$$\mathcal{J}(\mathbf{W}) \triangleq \sum_{p=1}^{P} \beta_p \left[\log\left(\det\left(\operatorname{diag}\left(\mathbf{W}\,\mathbf{R}_p\,\mathbf{W}^H\right)\right)\right) - \log\left(\det\left(\mathbf{W}\,\mathbf{R}_p\,\mathbf{W}^H\right)\right)\right], \tag{4}$$

where the matrices $\mathbf{R}_p \in \mathbb{C}^{N\times N}$ are Hermitian and need to be positive definite. The weights β_p are positive scalars. Consequently, the matrix products $\mathbf{W}\,\mathbf{R}_p\,\mathbf{W}^H \in$

$\mathbb{C}^{M\times M}$ are also Hermitian and positive definite, assuming $\mathbf{W}$ has full rank. The cost function (4) is motivated by the Hadamard inequality

$$\det(\,\mathbf{Q}\,) \leq \det(\,\mathrm{diag}(\,\mathbf{Q}\,)\,) \tag{5}$$

with equality if and only if $\mathbf{Q}$ is diagonal [9].

In the following, we will also use the cost function (4) as the basis of a Newton-type algorithm that we will derive. There are a few fundamental differences between the derived Newton algorithm and Pham's Jacobi-type algorithm. In Pham's algorithm, every iteration consists of a pairwise update of two rows at a time. One *sweep* of the algorithm consists of iterating once through all possible combinations of pairing two rows. Pham has shown that near a minimum, an iteration of his algorithm behaves similarly to a *quasi-Newton–Raphson* iteration.

In contrast to Pham's algorithm, we will use a pure Newton algorithm that updates *all coefficients* in $\mathbf{W}$ in every iteration. The main difficulty in deriving a pure Newton algorithm, as opposed to a *quasi*-Newton algorithm, is that the Hessian needs to be known in closed form in every iteration. One major difference to Pham's algorithm will be that we drop the constraint that $\mathbf{W}$ needs to be square, and we allow $\mathbf{W} \in \mathbb{C}^{M\times N}$ to be rectangular with $M \leq N$. Translated to the blind signal separation problem, our algorithm is also capable of working in cases where more sensor signals than source signals are present. This is particularly useful when the number of source signals is not known beforehand.

In the following sections, we will derive the gradient and Hessian of the cost function (4) in close form. To this end, we will use the matrix-form representation of the second-order Taylor series expansion as described by Manton in [13]. This form allows us to represent the gradient and the Hessian in a very compact form with the help of Kronecker products. As we will see, just as products of matrices reveal more structure than nested sums, the use of Kronecker products reveals the structure of the Hessian on an even higher level than by using matrices inside nested sums. Thorough treatments of Kronecker products and their properties are given in [3, 12].

1.1. Notation. The notation used throughout this paper is the following: Vectors are written in lowercase, matrices in uppercase. Matrix and vector transpose, complex conjugation, and Hermitian transpose are denoted by $(.)^T$, $(.)^*$, and $(.)^H$, respectively. The $M \times M$ identity matrix is denoted by $\mathbf{I}_{M\times M}$. The Frobenius norm and the trace of a matrix are denoted by $\|\,.\,\|_F$ and $\mathrm{tr}\,(.)$, respectively. The spectral radius of a matrix $\mathbf{Q}$ is the nonnegative real number $\rho(\mathbf{Q}) = \max\{|\lambda_{\max}(\mathbf{Q})|\}$; see [9]. Matrix dimensions are given in superscript, e.g., $\mathbf{W}^{M\times N}$. The operator $\mathrm{vec}(\,\mathbf{W}\,)$ forms a column vector by stacking the columns of $\mathbf{W}$, and $\mathbf{W}^{M\times N} = \mathrm{mat}_{M\times N}(\mathbf{w})$ is the inverse operation of $\mathbf{w} = \mathrm{vec}(\,\mathbf{W}^{M\times N}\,)$. The Kronecker product [3] is denoted by $\otimes$. The $MN \times MN$-dimensional permutation matrix $\mathbf{P}_{M\times N}$, where the subscript $M \times N$ is the argument of $\mathbf{P}_{M\times N}$, is uniquely defined with

$$\mathrm{vec}(\,\mathbf{W}^T\,) \equiv \mathbf{P}_{M\times N}\,\mathrm{vec}(\,\mathbf{W}^{M\times N}\,) \tag{6}$$

as $\mathrm{vec}(\,\mathbf{W}\,)$ and $\mathrm{vec}(\,\mathbf{W}^T\,)$ contain the same elements, just arranged in a different order. With $\bar{\mathbf{Q}} = \mathrm{diag}(\,\mathbf{q}\,)$ we get a square diagonal matrix that contains the elements of the vector $\mathbf{q}$ in its diagonal. The matrix $\bar{\mathbf{Q}} = \mathrm{diag}(\,\mathbf{Q}\,)$ is a diagonal matrix where its diagonal elements are the same as the diagonal elements of $\mathbf{Q}$, and

$$\mathrm{off}(\,\mathbf{Q}\,) \triangleq \mathbf{Q} - \mathrm{diag}(\,\mathbf{Q}\,) \tag{7}$$

keeps all off-diagonal elements of $\mathbf{Q}$ and sets all diagonal elements of $\mathbf{Q}$ to zero. Furthermore, we define the two $M^2 \times M^2$ diagonal projection matrices

$$\mathbf{P}_{\text{diag}} \triangleq \text{diag}(\,\text{vec}(\,\mathbf{I}_{M\times M}\,)\,), \tag{8}$$

$$\mathbf{P}_{\text{off}} \triangleq \mathbf{I}_{M^2\times M^2} - \mathbf{P}_{\text{diag}}, \tag{9}$$

which appear in the following two relations:

$$\text{vec}(\,\text{diag}(\,\mathbf{Z}\,)\,) = \mathbf{P}_{\text{diag}}\,\text{vec}(\,\mathbf{Z}\,), \tag{10}$$

$$\text{vec}(\,\text{off}(\,\mathbf{Z}\,)\,) = \mathbf{P}_{\text{off}}\,\text{vec}(\,\mathbf{Z}\,). \tag{11}$$

2. Second-order Taylor series of the cost function $\mathcal{J}(\mathbf{W})$.

2.1. Matrix form of second-order Taylor approximation. In the following we will derive the gradient and Hessian of the cost function (4) with respect to the free parameters, i.e., the elements of $\mathbf{W}$. First we need to define how the gradient and Hessian are represented. Since the free parameters in the cost function (4) are arranged in the matrix $\mathbf{W}$, we decide to use the matrix form of the second-order Taylor series as given by Manton in [13]:

Let $\mathcal{J} : \mathbb{C}^{M\times N} \to \mathbb{R}$ be a cost function. Then we can describe the Taylor series expansion of $\mathcal{J}$ at $\mathbf{W}$ as

$$\begin{aligned}\mathcal{J}(\mathbf{W}+\delta\mathbf{Z}) = \mathcal{J}(\mathbf{W}) &+ \delta\,\mathfrak{Re}\{\text{tr}\,(\mathbf{Z}^H\mathbf{D}_{\mathbf{W}})\}\\ &+ \frac{\delta^2}{2}\,\text{vec}(\,\mathbf{Z}\,)^H\mathbf{H}_{\mathbf{W}}\,\text{vec}(\,\mathbf{Z}\,)\\ &+ \frac{\delta^2}{2}\mathfrak{Re}\{\text{vec}(\,\mathbf{Z}\,)^T\mathbf{C}_{\mathbf{W}}\,\text{vec}(\,\mathbf{Z}\,)\} + O(\delta^3),\end{aligned} \tag{12}$$

where $\mathbf{W}, \mathbf{Z} \in \mathbb{C}^{M\times N}$, $\mathbf{D}_{\mathbf{W}} \in \mathbb{C}^{M\times N}$ is the gradient of $\mathcal{J}$ evaluated at $\mathbf{W}$, and $\mathbf{H}_{\mathbf{W}}, \mathbf{C}_{\mathbf{W}} \in \mathbb{C}^{MN\times MN}$ are the Hessian of $\mathcal{J}$ evaluated at $\mathbf{W}$. The scalar δ is a small real number. Uniqueness can be achieved by requiring $\mathbf{H}_{\mathbf{W}}^H = \mathbf{H}_{\mathbf{W}}$ and $\mathbf{C}_{\mathbf{W}}^T = \mathbf{C}_{\mathbf{W}}$.

In contrast to the commonly known vector form of the Taylor series expansion

$$\mathcal{J}(\mathbf{w}+\delta\mathbf{z}) = \mathcal{J}(\mathbf{w}) + \delta\,\mathbf{z}^T\mathbf{d} + \frac{\delta^2}{2}\mathbf{z}^T\mathbf{H}\,\mathbf{z} + O(\delta^3), \tag{13}$$

where the coefficients of $\mathbf{W}$ are rearranged in the real vector

$$\mathbf{w} \triangleq \begin{pmatrix}\mathbf{w}^{\text{re}}\\ \mathbf{w}^{\text{im}}\end{pmatrix} \triangleq \begin{pmatrix}\mathfrak{Re}\{\text{vec}(\mathbf{W})\}\\ \mathfrak{Im}\{\text{vec}(\mathbf{W})\}\end{pmatrix}, \tag{14}$$

the matrix form often reveals the structure of the gradient and the Hessian in a much more transparent form via matrix and Kronecker products. The gradients and the Hessians of the two Taylor expansion forms (12) and (13) can be transformed into each other, as described in [10].

2.2. Derivation of the gradient and Hessian. In order to simplify the derivation of the gradient and Hessian of $\mathcal{J}\,(.)$, we rewrite the cost function (4) as

$$\mathcal{J}\,(\mathbf{W}\,;\{\beta_p\},\{\mathbf{R}_p\}) \triangleq \sum_{p=1}^{P}\beta_p\,\widetilde{\mathcal{J}}\,(\mathbf{W};\mathbf{R}_p) \tag{15}$$

with

$$\widetilde{\mathcal{J}}\left(\mathbf{W};\mathbf{R}_p\right) \triangleq \mathcal{J}^{(1)}\left(\mathbf{W};\mathbf{R}_p\right) - \mathcal{J}^{(2)}\left(\mathbf{W};\mathbf{R}_p\right), \tag{16}$$

$$\mathcal{J}^{(1)}\left(\mathbf{W};\mathbf{R}\right) \triangleq \log\left(\det\left(\operatorname{diag}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{W}^H\right)\right)\right), \tag{17}$$

$$\mathcal{J}^{(2)}\left(\mathbf{W};\mathbf{R}\right) \triangleq \log\left(\det\left(\mathbf{W}\,\mathbf{R}\,\mathbf{W}^H\right)\right). \tag{18}$$

2.3. Gradient and Hessian of $\mathcal{J}^{(1)}(.)$. In order to derive the gradient and Hessian of $\mathcal{J}^{(1)}$, defined in (17), we do the following expansion:

$$\mathcal{J}^{(1)}\left(\mathbf{W}+\delta\,\mathbf{Z}\right) \triangleq \log\left(\det\left(\operatorname{diag}\left(\left(\mathbf{W}+\delta\,\mathbf{Z}\right)\mathbf{R}\left(\mathbf{W}+\delta\,\mathbf{Z}\right)^H\right)\right)\right) \tag{19}$$

$$= \log\left(\det\left(\operatorname{diag}\left(\left(\mathbf{W}\,\mathbf{R}\,\mathbf{W}^H + \delta\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H + \mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right) + \delta^2\mathbf{Z}\,\mathbf{R}\,\mathbf{Z}^H\right)\right)\right)\right). \tag{20}$$

By substituting

$$\bar{\mathbf{Q}} \triangleq \operatorname{diag}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{W}^H\right) \tag{21}$$

we can formulate (20) as

$$\begin{aligned}\mathcal{J}^{(1)}\left(\mathbf{W}+\delta\,\mathbf{Z}\right) &= \log\Big(\det\Big(\bar{\mathbf{Q}}\left(\mathbf{I}+\delta\,\bar{\mathbf{Q}}^{-1}\operatorname{diag}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H+\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right)\right.\\ &\qquad \left.+\,\delta^2\,\bar{\mathbf{Q}}^{-1}\operatorname{diag}\left(\mathbf{Z}\,\mathbf{R}\,\mathbf{Z}^H\right)\right)\Big)\Big) \qquad (22)\\ &= \mathcal{J}^{(1)}\left(\mathbf{W}\right) + \log\Big(\det\Big(\mathbf{I}+\delta\,\bar{\mathbf{Q}}^{-1}\operatorname{diag}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H+\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right)\\ &\qquad +\,\delta^2\,\bar{\mathbf{Q}}^{-1}\operatorname{diag}\left(\mathbf{Z}\,\mathbf{R}\,\mathbf{Z}^H\right)\Big)\Big) \qquad (23)\\ &= \mathcal{J}^{(1)}\left(\mathbf{W}\right) + \log\left(\det\left(\mathbf{I}+\delta\,\mathbf{A}+\delta^2\mathbf{B}\right)\right) \qquad (24)\end{aligned}$$

with

$$\mathbf{A} \triangleq \bar{\mathbf{Q}}^{-1}\operatorname{diag}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H+\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right), \tag{25}$$

$$\mathbf{B} \triangleq \bar{\mathbf{Q}}^{-1}\operatorname{diag}\left(\mathbf{Z}\,\mathbf{R}\,\mathbf{Z}^H\right). \tag{26}$$

Here we made use of $\det\mathbf{X}\mathbf{Y} = \det\mathbf{X}\det\mathbf{Y}$, which is valid for square matrices $\mathbf{X},\mathbf{Y}$. Before we continue to simplify (24), we introduce the following proposition.

PROPOSITION 2.1. *Let* $\mathbf{A},\mathbf{B}\in\mathbb{C}^{N\times N}$. *Then, for* $\delta\to 0$,

$$\log\left(\det\left(\mathbf{I}+\delta\,\mathbf{A}+\delta^2\mathbf{B}\right)\right) = \delta\operatorname{tr}\mathbf{A} + \delta^2\operatorname{tr}\mathbf{B} - \frac{1}{2}\delta^2\operatorname{tr}\mathbf{A}^2 + O(\delta^3). \tag{27}$$

Proof. Let $\mathbf{S}\in\mathbb{C}^{N\times N}$ with *spectral radius* $\rho(\mathbf{S})<1$. We know that for any nonsingular $\mathbf{G}\in\mathbb{C}^{N\times N}$ we have $\log\left(\det\left(\mathbf{G}\right)\right) = \operatorname{tr}\left(\log\left(\mathbf{G}\right)\right)$. We also know that $\log\left(\mathbf{I}+\mathbf{G}\right) = \sum_k \frac{(-1)^{k+1}}{k}\mathbf{G}^k$ for any $\mathbf{G}\in\mathbb{C}^{N\times N}$ with $\rho(\mathbf{G})<1$. Note that the assumption $\rho(\mathbf{S})<1$ implies that $\mathbf{I}+\mathbf{S}$ is nonsingular. Hence, combining the above results we have

$$\log(\det(\mathbf{I}+\mathbf{S})) = \sum_k \frac{(-1)^{k+1}}{k}\mathbf{S}^k. \tag{28}$$

Now, if we insert $\mathbf{S} = \delta\,\mathbf{A}+\delta^2\,\mathbf{B}$ into (28) and let $\delta\to 0$, we obtain (27). ∎

Note that the logarithm for complex arguments is not uniquely defined. However, we will only apply (28) for Hermitian matrices $\mathbf{S}$ with $\rho(\mathbf{S})<1$. Consequently all eigenvalues of $\mathbf{I}+\mathbf{S}$, and also $\det(\mathbf{I}+\mathbf{S})$, will be real and positive.

If we apply (27) in (24), we obtain

$$\mathcal{J}^{(1)}\left(\mathbf{W}+\delta\,\mathbf{Z}\right)=\mathcal{J}^{(1)}\left(\mathbf{W}\right)+\delta\,\mathrm{tr}\,\mathbf{A}+\delta^2\,\mathrm{tr}\,\mathbf{B}-\frac{1}{2}\delta^2\,\mathrm{tr}\,\mathbf{A}^2+O(\delta^3), \tag{29}$$

where $\mathbf{A}$ and $\mathbf{B}$ are defined in (25) and (26), respectively. We now analyze each of the three δ terms in (29) separately. We start with

$$\mathrm{tr}\,\mathbf{A}=\mathrm{tr}\left(\bar{\mathbf{Q}}^{-1}\,\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H+\bar{\mathbf{Q}}^{-1}\,\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right) \tag{30}$$

$$=\mathrm{tr}\left(\mathbf{Z}^H\,\bar{\mathbf{Q}}^{-1}\,\mathbf{W}\,\mathbf{R}+\mathbf{R}\,\mathbf{W}^H\,\bar{\mathbf{Q}}^{-1}\,\mathbf{Z}\right) \tag{31}$$

$$=2\,\mathfrak{Re}\big\{\,\mathrm{tr}\left(\mathbf{Z}^H\,\bar{\mathbf{Q}}^{-1}\,\mathbf{W}\,\mathbf{R}\right)\big\}. \tag{32}$$

Here we made use of $\mathrm{tr}\left(\mathrm{diag}(\mathbf{X})\,\mathrm{diag}(\mathbf{Y})\right)=\mathrm{tr}\left(\mathbf{X}\,\mathrm{diag}(\mathbf{Y})\right)=\mathrm{tr}\left(\mathrm{diag}(\mathbf{X})\,\mathbf{Y}\right)$. The third term in (29), where $\mathbf{A}$ is defined in (25), can be modified as

$$\begin{aligned}\mathrm{tr}\,\mathbf{A}^2&=\mathrm{tr}\left(\bar{\mathbf{Q}}^{-1}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H+\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right)\bar{\mathbf{Q}}^{-1}\,\mathrm{diag}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H+\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right)\right) &(33)\\
&=\mathrm{tr}\left(\bar{\mathbf{Q}}^{-1}\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H\bar{\mathbf{Q}}^{-1}\,\mathrm{diag}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H\right)\right)\\
&\quad+\mathrm{tr}\left(\bar{\mathbf{Q}}^{-1}\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H\bar{\mathbf{Q}}^{-1}\,\mathrm{diag}\left(\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right)\right)\\
&\quad+\mathrm{tr}\left(\bar{\mathbf{Q}}^{-1}\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\bar{\mathbf{Q}}^{-1}\,\mathrm{diag}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H\right)\right)\\
&\quad+\mathrm{tr}\left(\bar{\mathbf{Q}}^{-1}\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\bar{\mathbf{Q}}^{-1}\,\mathrm{diag}\left(\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right)\right) &(34)\\
&=2\,\mathrm{tr}\left(\mathbf{Z}^H\,\bar{\mathbf{Q}}^{-1}\,\mathrm{diag}\left(\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right)\bar{\mathbf{Q}}^{-1}\mathbf{W}\,\mathbf{R}\right)\\
&\quad+\mathrm{tr}\left(\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\,\bar{\mathbf{Q}}^{-1}\,\mathrm{diag}\left(\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right)\bar{\mathbf{Q}}^{-1}\right)\\
&\quad+\mathrm{tr}\left(\mathbf{Z}^H\,\bar{\mathbf{Q}}^{-1}\,\mathrm{diag}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H\right)\bar{\mathbf{Q}}^{-1}\mathbf{W}\,\mathbf{R}\right), &(35)\end{aligned}$$

where we used $\mathrm{tr}\left(\mathbf{X}\,\mathbf{Y}\right)=\mathrm{tr}\left(\mathbf{Y}\,\mathbf{X}\right)$ and some elementary properties of $\mathrm{tr}\left(.\right)$ with diagonal matrices. With the help of (94) and (95) we obtain

$$\begin{aligned}\mathrm{tr}\,\mathbf{A}^2&=2\,\mathrm{vec}(\mathbf{Z})^H\left[\left(\bar{\mathbf{Q}}^{-1}\,\mathbf{W}\,\mathbf{R}\right)^T\otimes\bar{\mathbf{Q}}^{-1}\right]\mathrm{vec}\big(\,\mathrm{diag}\left(\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right)\big)\\
&\quad+\mathrm{vec}(\mathbf{Z})^T\,\mathbf{P}_{M\times N}^T\left[\bar{\mathbf{Q}}^{-T}\otimes\mathbf{R}\,\mathbf{W}^H\,\bar{\mathbf{Q}}^{-1}\right]\mathrm{vec}\big(\,\mathrm{diag}\left(\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right)\big)\\
&\quad+\mathrm{vec}(\mathbf{Z})^H\left[\left(\bar{\mathbf{Q}}^{-1}\,\mathbf{W}\,\mathbf{R}\right)^T\otimes\bar{\mathbf{Q}}^{-1}\right]\mathrm{vec}\big(\,\mathrm{diag}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H\right)\big). &(36)\end{aligned}$$

Next we make use of (10). With further help of (93) and (84) we get

$$\begin{aligned}\mathrm{tr}\,\mathbf{A}^2&=2\,\mathrm{vec}(\mathbf{Z})^H\left[\left(\bar{\mathbf{Q}}^{-1}\,\mathbf{W}\,\mathbf{R}\right)^T\otimes\bar{\mathbf{Q}}^{-1}\right]\mathbf{P}_{\mathrm{diag}}\left[\left(\mathbf{R}\,\mathbf{W}^H\right)^T\otimes\mathbf{I}_M\right]\mathrm{vec}(\mathbf{Z})\\
&\quad+\mathrm{vec}(\mathbf{Z})^T\,\mathbf{P}_{M\times N}^T\left[\bar{\mathbf{Q}}^{-T}\otimes\mathbf{R}\,\mathbf{W}^H\,\bar{\mathbf{Q}}^{-1}\right]\mathbf{P}_{\mathrm{diag}}\left[\left(\mathbf{R}\,\mathbf{W}^H\right)^T\otimes\mathbf{I}_M\right]\mathrm{vec}(\mathbf{Z})\\
&\quad+\mathrm{vec}(\mathbf{Z})^H\left[\left(\bar{\mathbf{Q}}^{-1}\,\mathbf{W}\,\mathbf{R}\right)^T\otimes\bar{\mathbf{Q}}^{-1}\right]\mathbf{P}_{\mathrm{diag}}\left[\mathbf{I}_M\otimes\mathbf{W}\,\mathbf{R}\right]\mathbf{P}_{M\times N}\,\mathrm{vec}(\mathbf{Z})^*. &(37)\end{aligned}$$

In this case $\mathbf{P}_{\mathrm{diag}}$ is an $M^2\times M^2$-dimensional matrix. By making use of (92) we obtain

$$\begin{aligned}\mathrm{tr}\,\mathbf{A}^2&=2\,\mathrm{vec}(\mathbf{Z})^H\left[\mathbf{R}^T\,\mathbf{W}^T\,\bar{\mathbf{Q}}^{-T}\otimes\bar{\mathbf{Q}}^{-1}\right]\mathbf{P}_{\mathrm{diag}}\left[\mathbf{W}^*\,\mathbf{R}^T\otimes\mathbf{I}_M\right]\mathrm{vec}(\mathbf{Z})\\
&\quad+\mathrm{vec}(\mathbf{Z})^T\left[\mathbf{R}\,\mathbf{W}^H\,\bar{\mathbf{Q}}^{-1}\otimes\bar{\mathbf{Q}}^{-T}\right]\mathbf{P}_{M\times M}\,\mathbf{P}_{\mathrm{diag}}\left[\mathbf{W}^*\mathbf{R}^T\otimes\mathbf{I}_M\right]\mathrm{vec}(\mathbf{Z})\\
&\quad+\mathrm{vec}(\mathbf{Z})^H\left[\mathbf{R}^T\,\mathbf{W}^T\bar{\mathbf{Q}}^{-T}\otimes\bar{\mathbf{Q}}^{-1}\right]\mathbf{P}_{\mathrm{diag}}\,\mathbf{P}_{M\times M}\left[\mathbf{W}\,\mathbf{R}\otimes\mathbf{I}_M\right]\mathrm{vec}(\mathbf{Z})^*. &(38)\end{aligned}$$

Since $\mathbf{P}_{N\times M} = \mathbf{P}_{M\times N}^{-1} = \mathbf{P}_{M\times N}^{T}$ and $\mathbf{P}_{M\times M}\,\mathbf{P}_{\text{diag}} = \mathbf{P}_{\text{diag}}\,\mathbf{P}_{M\times M}$, we get

$$\begin{aligned}
\operatorname{tr}\mathbf{A}^2 &= 2\,\operatorname{vec}(\mathbf{Z})^H \left[\mathbf{R}^T\,\mathbf{W}^T\,\bar{\mathbf{Q}}^{-T}\otimes\bar{\mathbf{Q}}^{-1}\right]\mathbf{P}_{\text{diag}}\left[\mathbf{W}^*\,\mathbf{R}^T\otimes\mathbf{I}_M\right]\operatorname{vec}(\mathbf{Z}) \\
&\quad + \operatorname{vec}(\mathbf{Z})^T\left[\mathbf{R}\,\mathbf{W}^H\,\bar{\mathbf{Q}}^{-1}\otimes\bar{\mathbf{Q}}^{-T}\right]\mathbf{P}_{\text{diag}}\,\mathbf{P}_{M\times M}\left[\mathbf{W}^*\mathbf{R}^T\otimes\mathbf{I}_M\right]\operatorname{vec}(\mathbf{Z}) \\
&\quad + \operatorname{vec}(\mathbf{Z})^H\left[\mathbf{R}^T\,\mathbf{W}^T\bar{\mathbf{Q}}^{-T}\otimes\bar{\mathbf{Q}}^{-1}\right]\mathbf{P}_{\text{diag}}\,\mathbf{P}_{M\times M}\left[\mathbf{W}\,\mathbf{R}\otimes\mathbf{I}_M\right]\operatorname{vec}(\mathbf{Z})^* \qquad (39)\\
&= 2\,\operatorname{vec}(\mathbf{Z})^H\left[\mathbf{R}^T\,\mathbf{W}^T\,\bar{\mathbf{Q}}^{-T}\otimes\bar{\mathbf{Q}}^{-1}\right]\mathbf{P}_{\text{diag}}\left[\mathbf{W}^*\,\mathbf{R}^T\otimes\mathbf{I}_M\right]\operatorname{vec}(\mathbf{Z}) \\
&\quad + 2\,\mathfrak{Re}\big\{\operatorname{vec}(\mathbf{Z})^T\left[\mathbf{R}\,\mathbf{W}^H\,\bar{\mathbf{Q}}^{-1}\otimes\bar{\mathbf{Q}}^{-T}\right]\mathbf{P}_{\text{diag}}\,\mathbf{P}_{M\times M}\left[\mathbf{W}^*\mathbf{R}^T\otimes\mathbf{I}_M\right]\operatorname{vec}(\mathbf{Z})\big\}. \qquad (40)
\end{aligned}$$

In the last step we used the fact that $\mathbf{R}^* = \mathbf{R}^T$ and $\bar{\mathbf{Q}}^* = \bar{\mathbf{Q}}^T$, as $\mathbf{R}$ and $\bar{\mathbf{Q}}$ are both Hermitian. Finally, the second term in (29) can be modified with (95) as

$$\begin{aligned}
\operatorname{tr}\mathbf{B} &= \operatorname{tr}\left(\bar{\mathbf{Q}}^{-1}\,\operatorname{diag}\big(\mathbf{Z}\,\mathbf{R}\,\mathbf{Z}^H\big)\right) = \operatorname{tr}\left(\bar{\mathbf{Q}}^{-1}\,\mathbf{Z}\,\mathbf{R}\,\mathbf{Z}^H\right) \qquad (41)\\
&= \operatorname{tr}\left(\mathbf{Z}^H\bar{\mathbf{Q}}^{-1}\,\mathbf{Z}\,\mathbf{R}\right) \qquad (42)\\
&= \operatorname{vec}(\mathbf{Z})^H\left[\mathbf{R}^T\otimes\bar{\mathbf{Q}}^{-1}\right]\operatorname{vec}(\mathbf{Z}). \qquad (43)
\end{aligned}$$

Inserting (32), (40), and (43) into (29) yields

$$\begin{aligned}
&\mathcal{J}^{(1)}(\mathbf{W}+\delta\,\mathbf{Z}) \\
&\quad = \mathcal{J}^{(1)}(\mathbf{W}) + 2\,\delta\,\mathfrak{Re}\big\{\operatorname{tr}\left(\mathbf{Z}^H\,\bar{\mathbf{Q}}^{-1}\,\mathbf{W}\,\mathbf{R}\right)\big\} + \delta^2\operatorname{vec}(\mathbf{Z})^H\cdot \\
&\qquad \cdot\Big(\left[\mathbf{R}^T\otimes\bar{\mathbf{Q}}^{-1}\right] - \left[\mathbf{R}^T\,\mathbf{W}^T\,\bar{\mathbf{Q}}^{-T}\otimes\bar{\mathbf{Q}}^{-1}\right]\mathbf{P}_{\text{diag}}\left[\mathbf{W}^*\,\mathbf{R}^T\otimes\mathbf{I}_M\right]\Big)\operatorname{vec}(\mathbf{Z}) \\
&\qquad - \delta^2\,\mathfrak{Re}\big\{\operatorname{vec}(\mathbf{Z})^T\left[\mathbf{R}\,\mathbf{W}^H\,\bar{\mathbf{Q}}^{-1}\otimes\bar{\mathbf{Q}}^{-T}\right]\mathbf{P}_{\text{diag}}\,\mathbf{P}_{M\times M}\left[\mathbf{W}^*\mathbf{R}^T\otimes\mathbf{I}_M\right]\operatorname{vec}(\mathbf{Z})\big\} \\
&\quad + O(\delta^3). \qquad (44)
\end{aligned}$$

By coefficient comparison between (44) and the matrix form of the second-order Taylor series (12), and using $\mathbf{R}^T = \mathbf{R}^*$ and (91), we finally obtain

$$\mathbf{D}_{\mathbf{W}}^{(1)} = 2\,\bar{\mathbf{Q}}^{-1}\,\mathbf{W}\,\mathbf{R}, \qquad (45)$$

$$\mathbf{H}_{\mathbf{W}}^{(1)} = 2\left[\mathbf{R}^T\otimes\bar{\mathbf{Q}}^{-1}\right] - 2\left[\mathbf{R}^T\,\mathbf{W}^T\otimes\mathbf{I}_M\right]\left[\bar{\mathbf{Q}}^{-T}\otimes\bar{\mathbf{Q}}^{-1}\right]\mathbf{P}_{\text{diag}}\left[\mathbf{W}^*\,\mathbf{R}^*\otimes\mathbf{I}_M\right], \qquad (46)$$

$$\mathbf{C}_{\mathbf{W}}^{(1)} = -2\left[\mathbf{R}\,\mathbf{W}^H\otimes\mathbf{I}_M\right]\left[\bar{\mathbf{Q}}^{-1}\otimes\bar{\mathbf{Q}}^{-T}\right]\mathbf{P}_{\text{diag}}\,\mathbf{P}_{M\times M}\left[\mathbf{W}^*\mathbf{R}^*\otimes\mathbf{I}_M\right], \qquad (47)$$

where $\bar{\mathbf{Q}}$ is defined in (21).

2.4. Gradient and Hessian of $\mathcal{J}^{(2)}(.)$. Deriving the gradient and Hessian of $\mathcal{J}^{(2)}(\mathbf{W}) \triangleq \log\left(\det\left(\mathbf{W}\,\mathbf{R}\,\mathbf{W}^H\right)\right)$, as defined in (18), can be done in a similar way to how it was done for $\mathcal{J}^{(1)}$ in the previous section. To this end, we expand $\mathcal{J}^{(2)}$ as

$$\begin{aligned}
&\mathcal{J}^{(2)}(\mathbf{W}+\delta\,\mathbf{Z}) = \log\left(\det\left((\mathbf{W}+\delta\,\mathbf{Z})\,\mathbf{R}\,(\mathbf{W}+\delta\,\mathbf{Z})^H\right)\right) \qquad (48)\\
&\qquad = \log\left(\det\left(\mathbf{W}\,\mathbf{R}\,\mathbf{W}^H + \delta\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H + \mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right) + \delta^2\mathbf{Z}\,\mathbf{R}\,\mathbf{Z}^H\right)\right). \qquad (49)
\end{aligned}$$

By substituting

$$\mathbf{Q} \triangleq \mathbf{W}\,\mathbf{R}\,\mathbf{W}^H \qquad (50)$$

we can rewrite (49) as

$$\begin{aligned}
&\mathcal{J}^{(2)}(\mathbf{W}+\delta\,\mathbf{Z}) \\
&\quad = \log\left(\det\left(\mathbf{Q}\,\left(\mathbf{I} + \delta\,\mathbf{Q}^{-1}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H + \mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right) + \delta^2\,\mathbf{Q}^{-1}\mathbf{Z}\,\mathbf{R}\,\mathbf{Z}^H\right)\right)\right) \qquad (51)\\
&\quad = \mathcal{J}^{(2)}(\mathbf{W}) + \log\left(\det\left(\mathbf{I} + \delta\,\mathbf{Q}^{-1}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H + \mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right) + \delta^2\,\mathbf{Q}^{-1}\mathbf{Z}\,\mathbf{R}\,\mathbf{Z}^H\right)\right). \qquad (52)
\end{aligned}$$

By inspection, we see that (52) has the same structure as (23) and therefore can also be written in the form

$$\mathcal{J}^{(2)}\left(\mathbf{W}+\delta\,\mathbf{Z}\right)=\mathcal{J}^{(2)}\left(\mathbf{W}\right)+\log\left(\det\left(\mathbf{I}+\delta\,\mathbf{A}+\delta^2\mathbf{B}\right)\right), \tag{53}$$

this time with

$$\mathbf{A}\triangleq\mathbf{Q}^{-1}\left(\mathbf{W}\,\mathbf{R}\,\mathbf{Z}^H+\mathbf{Z}\,\mathbf{R}\,\mathbf{W}^H\right), \tag{54}$$

$$\mathbf{B}\triangleq\mathbf{Q}^{-1}\mathbf{Z}\,\mathbf{R}\,\mathbf{Z}^H. \tag{55}$$

One difference, though, is that the matrices $\mathbf{Q}$, $\mathbf{A}$, and $\mathbf{B}$ are this time no longer diagonal, in general. However, since in the previous calculations we only made the assumption that these matrices needed to be Hermitian and not diagonal, we can use (27) again and carry out the further steps in exactly the same manner as it was done for $\mathcal{J}^{(1)}$. By going through the same derivation steps, we finally obtain

$$\mathbf{D}_{\mathbf{W}}^{(2)}=2\,\mathbf{Q}^{-1}\,\mathbf{W}\,\mathbf{R}, \tag{56}$$

$$\mathbf{H}_{\mathbf{W}}^{(2)}=2\left[\,\mathbf{R}^T\otimes\mathbf{Q}^{-1}\right]-2\left[\,\mathbf{R}^T\,\mathbf{W}^T\otimes\mathbf{I}_M\,\right]\left[\,\mathbf{Q}^{-T}\otimes\mathbf{Q}^{-1}\right]\left[\,\mathbf{W}^*\,\mathbf{R}^*\otimes\mathbf{I}_M\right], \tag{57}$$

$$\mathbf{C}_{\mathbf{W}}^{(2)}=-2\left[\,\mathbf{R}\,\mathbf{W}^H\otimes\mathbf{I}_M\right]\left[\,\mathbf{Q}^{-1}\otimes\mathbf{Q}^{-T}\right]\mathbf{P}_{M\times M}\left[\,\mathbf{W}^*\mathbf{R}^*\otimes\mathbf{I}_M\right]. \tag{58}$$

By comparing (56)–(58) with (45)–(47), respectively, basically $\bar{\mathbf{Q}}$ is replaced by its nondiagonal version $\mathbf{Q}$, and $\mathbf{P}_{\text{diag}}$ is replaced by $\mathbf{I}_M$.

2.5. Gradient and Hessian of $\mathcal{J}\,(.)$. Because of the definitions (16), (17), and (18) we can write the gradient and the Hessian of $\widetilde{\mathcal{J}}(.)$ as

$$\widetilde{\mathbf{D}}_{\mathbf{W}}\left(\mathbf{W};\mathbf{R}_p\right)=\mathbf{D}_{\mathbf{W}}^{(1)}\left(\mathbf{W};\mathbf{R}_p\right)-\mathbf{D}_{\mathbf{W}}^{(2)}\left(\mathbf{W};\mathbf{R}_p\right), \tag{59}$$

$$\widetilde{\mathbf{H}}_{\mathbf{W}}\left(\mathbf{W};\mathbf{R}_p\right)=\mathbf{H}_{\mathbf{W}}^{(1)}\left(\mathbf{W};\mathbf{R}_p\right)-\mathbf{H}_{\mathbf{W}}^{(2)}\left(\mathbf{W};\mathbf{R}_p\right), \tag{60}$$

$$\widetilde{\mathbf{C}}_{\mathbf{W}}\left(\mathbf{W};\mathbf{R}_p\right)=\mathbf{C}_{\mathbf{W}}^{(1)}\left(\mathbf{W};\mathbf{R}_p\right)-\mathbf{C}_{\mathbf{W}}^{(2)}\left(\mathbf{W};\mathbf{R}_p\right), \tag{61}$$

where $\mathbf{D}_{\mathbf{W}}^{(1)}$, $\mathbf{H}_{\mathbf{W}}^{(1)}$, and $\mathbf{C}_{\mathbf{W}}^{(1)}$ are defined in (45), (46), and (47), respectively, and $\mathbf{D}_{\mathbf{W}}^{(2)}$, $\mathbf{H}_{\mathbf{W}}^{(2)}$, and $\mathbf{C}_{\mathbf{W}}^{(2)}$ are defined in (56), (57), and (58), respectively. By inserting these terms into (59), (60), and (61), after some rearranging we finally obtain, for all p,

$$\widetilde{\mathbf{D}}_{\mathbf{W}}\left(\mathbf{W};\mathbf{R}\right)=2\left(\,\bar{\mathbf{Q}}^{-1}-\mathbf{Q}^{-1}\,\right)\mathbf{W}\,\mathbf{R}, \tag{62}$$

$$\begin{aligned}\widetilde{\mathbf{H}}_{\mathbf{W}}\left(\mathbf{W};\mathbf{R}\right)=\;&2\left[\,\mathbf{R}^T\otimes\left(\,\bar{\mathbf{Q}}^{-1}-\mathbf{Q}^{-1}\,\right)\right]\\&+2\left[\,\mathbf{R}^T\,\mathbf{W}^T\otimes\mathbf{I}_M\,\right]\Big(\left[\,\mathbf{Q}^{-T}\otimes\mathbf{Q}^{-1}\,\right]-\left[\,\bar{\mathbf{Q}}^{-T}\otimes\bar{\mathbf{Q}}^{-1}\,\right]\mathbf{P}_{\text{diag}}\Big)\\&\cdot\left[\,\mathbf{W}^*\,\mathbf{R}^*\otimes\mathbf{I}_M\right],\end{aligned} \tag{63}$$

$$\widetilde{\mathbf{C}}_{\mathbf{W}}\left(\mathbf{W};\mathbf{R}\right)=2\left[\,\mathbf{R}\,\mathbf{W}^H\otimes\mathbf{I}_M\,\right]\Big(\left[\,\mathbf{Q}^{-1}\otimes\mathbf{Q}^{-T}\,\right]-\left[\,\bar{\mathbf{Q}}^{-1}\otimes\bar{\mathbf{Q}}^{-T}\,\right]\mathbf{P}_{\text{diag}}\Big)\cdot\mathbf{P}_{M\times M}\left[\,\mathbf{W}^*\mathbf{R}^*\otimes\mathbf{I}_M\right]. \tag{64}$$

Finally, because of (15), we can write the gradient and the Hessian of the cost function $\mathcal{J}$ as

$$\mathbf{D}_{\mathbf{W}}=\sum_{p=1}^{P}\beta_p\,\widetilde{\mathbf{D}}_{\mathbf{W}}\left(\mathbf{W};\mathbf{R}_p\right), \tag{65}$$

$$\mathbf{H}_{\mathbf{W}} = \sum_{p=1}^{P} \beta_p \, \widetilde{\mathbf{H}}_{\mathbf{W}} \left(\mathbf{W}; \mathbf{R}_p \right), \tag{66}$$

$$\mathbf{C}_{\mathbf{W}} = \sum_{p=1}^{P} \beta_p \, \widetilde{\mathbf{C}}_{\mathbf{W}} \left(\mathbf{W}; \mathbf{R}_p \right). \tag{67}$$

Equations (62) to (67) are the gradient and the Hessian of the cost function (4).

2.6. Comparison between gradient and Hessian of $\mathcal{J}$ and $\mathcal{J}_{\text{off}}$. A well-known cost function used for the joint diagonalization problem is [5]

$$\mathcal{J}_{\text{off}} \left(\mathbf{W} \, ; \{\beta_p\}, \{\mathbf{R}_p\} \right) \triangleq \sum_{p=1}^{P} \beta_p \left\| \, \text{off}(\, \mathbf{W} \, \mathbf{R}_p \, \mathbf{W}^H \,) \, \right\|_F^2 \tag{68}$$

subject to a constraint that prevents $\mathbf{W}$ from becoming zero. We now like to make a comparison between the gradient and Hessian of $\mathcal{J}$ and $\mathcal{J}_{\text{off}}$. The gradient and Hessian of the term $\hat{\mathcal{J}} \triangleq \|\text{off}(\, \mathbf{W} \, \mathbf{R}_p \, \mathbf{W}^H \,)\|_F^2$, which appears in (68), are [10]

$$\widehat{\mathbf{D}}_{\mathbf{W}} \left(\mathbf{W}; \mathbf{R} \right) = 4 \, \text{off}(\, \mathbf{Q} \,) \, \mathbf{W} \, \mathbf{R}, \tag{69}$$

$$\widehat{\mathbf{H}}_{\mathbf{W}} \left(\mathbf{W}; \mathbf{R} \right) = 4 \left[\, \mathbf{R}^T \otimes \text{off}(\, \mathbf{Q} \,) \, \right] + 4 \left[\, \mathbf{R}^T \, \mathbf{W}^T \otimes \mathbf{I}_M \, \right] \mathbf{P}_{\text{off}} \left[\, \mathbf{W}^* \, \mathbf{R}^* \otimes \mathbf{I}_M \right], \tag{70}$$

$$\widehat{\mathbf{C}}_{\mathbf{W}} \left(\mathbf{W}; \mathbf{R} \right) = 4 \left[\, \mathbf{R} \, \mathbf{W}^H \otimes \mathbf{I}_M \, \right] \mathbf{P}_{\text{off}} \, \mathbf{P}_{M \times M} \left[\, \mathbf{W}^* \mathbf{R}^* \otimes \mathbf{I}_M \right]. \tag{71}$$

We wish to bring the gradient and Hessian of $\mathcal{J}$ into a similar form. To this end, we reformulate the term $\bar{\mathbf{Q}}^{-1} - \mathbf{Q}^{-1}$ as

$$\bar{\mathbf{Q}}^{-1} - \mathbf{Q}^{-1} = \bar{\mathbf{Q}}^{-1} \, (\, \mathbf{Q} - \bar{\mathbf{Q}} \,) \mathbf{Q}^{-1} = \bar{\mathbf{Q}}^{-1} \, \text{off}(\, \mathbf{Q} \,) \, \mathbf{Q}^{-1}, \tag{72}$$

where off(.) is defined in (7) and $\bar{\mathbf{Q}} \triangleq \text{diag}(\, \mathbf{Q} \,)$; see (21) and (50). Furthermore, since $\mathbf{P}_{\text{diag}} + \mathbf{P}_{\text{off}} = \mathbf{I}$ we have

$$\begin{aligned} &\left[\, \mathbf{Q}^{-T} \otimes \mathbf{Q}^{-1} \, \right] - \left[\, \bar{\mathbf{Q}}^{-T} \otimes \bar{\mathbf{Q}}^{-1} \, \right] \mathbf{P}_{\text{diag}} \\ &\qquad = \left(\left[\, \mathbf{Q}^{-T} \otimes \mathbf{Q}^{-1} \, \right] - \left[\, \bar{\mathbf{Q}}^{-T} \otimes \bar{\mathbf{Q}}^{-1} \, \right] \right) \mathbf{P}_{\text{diag}} + \left[\, \mathbf{Q}^{-T} \otimes \mathbf{Q}^{-1} \, \right] \mathbf{P}_{\text{off}}. \end{aligned} \tag{73}$$

By inserting (72) into (62) and (63), and (73) into (63) and (64) we can write the gradient and Hessian of $\mathcal{J}$ as

$$\widetilde{\mathbf{D}}_{\mathbf{W}} \left(\mathbf{W}; \mathbf{R} \right) = 2 \, \bar{\mathbf{Q}}^{-1} \, \text{off}(\, \mathbf{Q} \,) \, \mathbf{Q}^{-1} \, \mathbf{W} \, \mathbf{R}, \tag{74}$$

$$\begin{aligned} \widetilde{\mathbf{H}}_{\mathbf{W}} \left(\mathbf{W}; \mathbf{R} \right) &= 2 \left[\, \mathbf{R}^T \otimes \bar{\mathbf{Q}}^{-1} \, \text{off}(\, \mathbf{Q} \,) \, \mathbf{Q}^{-1} \, \right] \\ &\quad + 2 \left[\, \mathbf{R}^T \, \mathbf{W}^T \otimes \mathbf{I}_M \, \right] \left(\left[\, \mathbf{Q}^{-T} \otimes \mathbf{Q}^{-1} \, \right] - \left[\, \bar{\mathbf{Q}}^{-T} \otimes \bar{\mathbf{Q}}^{-1} \, \right] \right) \mathbf{P}_{\text{diag}} \left[\, \mathbf{W}^* \, \mathbf{R}^* \otimes \mathbf{I}_M \right] \\ &\quad + 2 \left[\, \mathbf{R}^T \, \mathbf{W}^T \otimes \mathbf{I}_M \, \right] \left[\, \mathbf{Q}^{-T} \otimes \mathbf{Q}^{-1} \, \right] \mathbf{P}_{\text{off}} \left[\, \mathbf{W}^* \, \mathbf{R}^* \otimes \mathbf{I}_M \right], \end{aligned} \tag{75}$$

$$\begin{aligned} &\widetilde{\mathbf{C}}_{\mathbf{W}} \left(\mathbf{W}; \mathbf{R} \right) \\ &\quad = 2 \left[\, \mathbf{R} \, \mathbf{W}^H \otimes \mathbf{I}_M \, \right] \left(\left[\, \mathbf{Q}^{-1} \otimes \mathbf{Q}^{-T} \, \right] - \left[\, \bar{\mathbf{Q}}^{-1} \otimes \bar{\mathbf{Q}}^{-T} \, \right] \right) \mathbf{P}_{\text{diag}} \, \mathbf{P}_{M \times M} \left[\, \mathbf{W}^* \, \mathbf{R}^* \otimes \mathbf{I}_M \right] \\ &\quad + 2 \left[\, \mathbf{R} \, \mathbf{W}^H \otimes \mathbf{I}_M \, \right] \left[\, \mathbf{Q}^{-1} \otimes \mathbf{Q}^{-T} \, \right] \mathbf{P}_{\text{off}} \, \mathbf{P}_{M \times M} \left[\, \mathbf{W}^* \, \mathbf{R}^* \otimes \mathbf{I}_M \right]. \end{aligned} \tag{76}$$

This form allows us to see clear similarities between the gradient and Hessian terms of the two cost functions $\mathcal{J}_{\text{off}}$ and $\mathcal{J}$ when comparing (69)–(71) with (74)–(76), respectively: In the gradient and the first term of the Hessian, off($\mathbf{Q}$) is replaced by

$\bar{\mathbf{Q}}^{-1}\,\mathrm{off}(\,\mathbf{Q}\,)\,\mathbf{Q}^{-1}$. Note that $\mathbf{Q}$ and $\bar{\mathbf{Q}}$ are both positive definite matrices. The last terms of the Hessian terms also show a very similar structure. The only difference is the additional term $\mathbf{Q}^{-T} \otimes \mathbf{Q}^{-1}$ in the Hessian of $\mathcal{J}$, which is also positive definite. The second term of $\widetilde{\mathbf{H}}_{\mathbf{W}}$ and the first term of $\widetilde{\mathbf{C}}_{\mathbf{W}}$ have no corresponding terms in $\widehat{\mathbf{H}}_{\mathbf{W}}$ and $\widehat{\mathbf{C}}_{\mathbf{W}}$.

2.7. Discussion of critical points. The critical points are defined where the gradient of $\mathcal{J}$ becomes zero. From (74) we see that $\widetilde{\mathbf{D}}_{\mathbf{W}}$ becomes zero if $\mathrm{off}(\,\mathbf{Q}\,) \equiv \mathbf{0}$ and consequently $\mathbf{Q} \equiv \bar{\mathbf{Q}}$. Since we can reformulate, similar to (72),

(77)
$$\left[\,\mathbf{Q}^{-T} \otimes \mathbf{Q}^{-1}\,\right] - \left[\,\bar{\mathbf{Q}}^{-T} \otimes \bar{\mathbf{Q}}^{-1}\,\right] = -\left[\,\mathbf{Q}^{-T} \otimes \mathbf{Q}^{-1}\,\right] \mathrm{off}(\;\mathbf{Q}^{T} \otimes \mathbf{Q}\;)\left[\,\bar{\mathbf{Q}}^{-T} \otimes \bar{\mathbf{Q}}^{-1}\,\right]$$

and the Kronecker product of two diagonal matrices is a diagonal matrix, the term $\mathrm{off}(\,\bar{\mathbf{Q}}^{T} \otimes \bar{\mathbf{Q}}\,)$ becomes zero. Hence, the Hessian terms of $\mathcal{J}$ at the critical points are

$$\widetilde{\mathbf{H}}_{\mathbf{W}}\,(\mathbf{W};\mathbf{R}) = 2\left[\,\mathbf{R}^{T}\,\mathbf{W}^{T} \otimes \mathbf{I}_{M}\,\right]\left[\,\bar{\mathbf{Q}}^{-T} \otimes \bar{\mathbf{Q}}^{-1}\,\right]\mathbf{P}_{\mathrm{off}}\left[\,\mathbf{W}^{*}\,\mathbf{R}^{*} \otimes \mathbf{I}_{M}\right], \tag{78}$$

$$\widetilde{\mathbf{C}}_{\mathbf{W}}\,(\mathbf{W};\mathbf{R}) = 2\left[\,\mathbf{R}\,\mathbf{W}^{H} \otimes \mathbf{I}_{M}\,\right]\left[\,\bar{\mathbf{Q}}^{-1} \otimes \bar{\mathbf{Q}}^{-T}\,\right]\mathbf{P}_{\mathrm{off}}\,\mathbf{P}_{M\times M}\left[\,\mathbf{W}^{*}\,\mathbf{R}^{*} \otimes \mathbf{I}_{M}\right]. \tag{79}$$

When comparing (78) and (79) with (70) and (71), respectively, we make the interesting discovery that, besides the diagonal matrix $\bar{\mathbf{Q}}^{-T} \otimes \bar{\mathbf{Q}}^{-1}$ and a scaling factor, the Hessian of $\mathcal{J}$ and $\mathcal{J}_{\mathrm{off}}$ at the critical points have an identical structure.

3. The Newton algorithm. Once we have derived the gradient and Hessian of our cost function, we can now formulate the Newton algorithm. The Newton update at iteration k can be written as

$$\mathbf{W}_{k+1} = \mathbf{W}_{k} + \mu_{k}\mathbf{S}_{k}, \tag{80}$$

where $\mathbf{S}_k$ is the *search direction* and μ_k is the *step size* of the kth update. The individual steps of the Newton algorithm are given in Figure 1. Since our cost function is nonquadratic, we use a modified Newton step that incorporates an Armijo line search. In the vicinity of the minimum the update will approach the pure Newton step.

The described Newton algorithm is built upon a vector-form Newton algorithm, where the complex coefficients of $\mathbf{W}_k$ are arranged in a length $2MN$ real vector $\mathbf{w}_k$ as defined in (14). Hence, the vector $\mathbf{d}_k$ and the matrix $\mathbf{H}_k$ are the gradient and Hessian of the vector form of the Taylor series expansion of $\mathcal{J}(\mathbf{w}_k)$; see (13).

Since our cost function is not quadratic, the Hessian $\mathbf{H}_k$ can have negative eigenvalues. By choosing σ_k such that $\mathbf{H}_k + \sigma_k\mathbf{I}$ becomes positive definite, the inverse $[\,\mathbf{H}_k + \sigma_k\mathbf{I}]^{-1}$ will be positive definite as well. This will ensure that $\mathbf{s}_k$ and $\mathbf{S}_k$ will point to a descent direction, just like the negative gradient $-\mathbf{d}_k$ does. Hence, σ_k must be chosen larger than $-\lambda_{min}(\mathbf{H}_k)$ if $\mathbf{H}_k$ has nonpositive eigenvalues, where λ_{min} is the smallest eigenvalue of $\mathbf{H}_k$. In the vicinity of a local minimum, σ_k will become zero and the update will approach the pure Newton update for $\mu_k = 1$. On the other hand, if σ_k is chosen very large, the search direction $\mathbf{s}_k$ will become close to the direction of the negative gradient. For efficiency reasons, $\mathbf{H}_k$ is often regularized via a modified Cholesky factorization method [2, sec. 1.4]. Therefore the described modified Newton algorithm should be understood more as a prototype algorithm. The step size μ_k is obtained from a line-search step, e.g., an *Armijo line-search method* [2, 6, 11, 15], which guarantees that $\mathcal{J}(\mathbf{W}_{k+1}) \leq \mathcal{J}(\mathbf{W}_k)$ and μ_k is chosen *not too small*. The reason to include a variable step size μ_k into the Newton update is motivated by the

Modified Newton algorithm

Initialization ($k = 0$): $\mathbf{W}_0 \in \mathbb{C}^{M\times N}$.

For $k = 1, 2, \ldots$ until convergence,

1. Compute gradient $\mathbf{D}_{\mathbf{W}k}$ with (65) and (62).
2. Compute Hessian $\{\mathbf{H}_{\mathbf{W}k}, \mathbf{C}_{\mathbf{W}k}\}$ with (66), (63), (67), and (64).
3. Compute the real vector
$$\mathbf{d}_k \triangleq \begin{pmatrix} \mathbf{d}^{\mathrm{re}} \\ \mathbf{d}^{\mathrm{im}} \end{pmatrix} := \begin{pmatrix} \mathfrak{Re}\{\mathrm{vec}(\mathbf{D}_{\mathbf{W}})\} \\ \mathfrak{Im}\{\mathrm{vec}(\mathbf{D}_{\mathbf{W}})\} \end{pmatrix}.$$
4. Compute the real matrix
$$\mathbf{H}_k := \begin{bmatrix} \mathfrak{Re}\{\mathbf{H}_{\mathbf{W}} + \mathbf{C}_{\mathbf{W}}\} & -\mathfrak{Im}\{\mathbf{H}_{\mathbf{W}} + \mathbf{C}_{\mathbf{W}}\} \\ \mathfrak{Im}\{\mathbf{H}_{\mathbf{W}} - \mathbf{C}_{\mathbf{W}}\} & \mathfrak{Re}\{\mathbf{H}_{\mathbf{W}} - \mathbf{C}_{\mathbf{W}}\} \end{bmatrix}.$$
5. Compute the real vector
$$\mathbf{s}_k := -\left[\mathbf{H}_k + \sigma_k \mathbf{I}\right]^{-1} \mathbf{d}_k,$$
where $\sigma_k \geq 0$ is chosen such that $\mathbf{H}_k + \sigma_k \mathbf{I}$ becomes positive definite.
6. Compute the complex matrix $\mathbf{S}_k := \mathrm{mat}_{M\times N}(\mathbf{s}_k^{\mathrm{re}} + j\mathbf{s}_k^{\mathrm{im}})$ which corresponds to the inverse operation of
$$\mathbf{s}_k \triangleq \begin{pmatrix} \mathbf{s}^{\mathrm{re}} \\ \mathbf{s}^{\mathrm{im}} \end{pmatrix} = \begin{pmatrix} \mathfrak{Re}\{\mathrm{vec}(\mathbf{S}_k)\} \\ \mathfrak{Im}\{\mathrm{vec}(\mathbf{S}_k)\} \end{pmatrix}.$$
7. Perform update $\mathbf{W}_{k+1} := \mathbf{W}_k + \mu_k \mathbf{S}_k$, where the step size μ_k is found via a line-search algorithm, e.g., an Armijo line search (see Appendix B).

FIG. 1. *Modified Newton algorithm.*

fact that in our case $\mathcal{J}$ is a nonquadratic cost function. In this case the Newton step, $\mu_k = 1$, often overshoots the local minimum, even if the Hessian $\mathbf{H}_k$ is positive definite. Close to a minimum, where the cost function can be approximated by a quadratic curvature, the Armijo line-search method will often set μ_k automatically to one.

4. Simulation example. We give now a simulation example where the performance between the Newton algorithm and a steepest-descent algorithm are compared. This simulation is set up such that a perfect joint diagonalization is possible. The chosen parameters are $M = 3$ and $N = 5$. For each simulation trial we generate a random complex matrix $\mathbf{A} \in \mathbb{C}^{5\times 3}$, such that $\mathbf{A}^H\mathbf{A} = \mathbf{I}_3$. Then we generate for each trial a set of $P = 15$ correlation matrices $\{\mathbf{R}_p\}_{p=1}^{15} = \{\mathbf{A}\,\mathbf{\Lambda}_p\,\mathbf{A}^H\}_{p=1}^{15}$ where each $\mathbf{\Lambda}_p \in \mathbb{R}^{3\times 3}$ is a diagonal matrix whose elements are randomly chosen from a uniform distribution between 0.1 and 1. Hence, each $\mathbf{\Lambda}_p$ is positive definite, and each $\mathbf{R}_p$ is positive semidefinite and has rank 3. Figure 2 compares the performance between the Newton algorithm, using Armijo line searches, and a gradient-type update. The top curve shows the performance of the cost function $\mathcal{J}$, defined in (4); the bottom curve

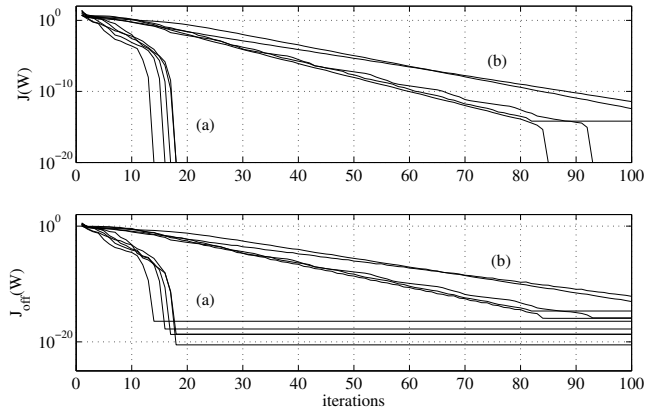


FIG. 2. *Learning curves of* $\mathcal{J}(\mathbf{W}_k)$ *(top) and* $\mathcal{J}_{\text{off}}(\mathbf{W}_k)$ *(bottom) for five independent simulations using* (a) *the modified Newton method and* (b) *the steepest-descent algorithm.*

shows how the cost function $\mathcal{J}_{\text{off}}$, given in (68), behaves. For this simulation we premultiply $\mathbf{W}_k$ after every iteration with a diagonal matrix, such that the rows of $\mathbf{W}_k$ have unit norm, i.e., $\text{diag}(\mathbf{W}_k\,\mathbf{W}_k^H) = \mathbf{I}_3$. This normalization step does not affect the value of the cost function $\mathcal{J}$, as $\mathcal{J}$ is scale-invariant to such an operation; however, it is important for a meaningful interpretation of $\mathcal{J}_{\text{off}}$. Without this normalization a small value of $\mathcal{J}_{\text{off}}$ could also be caused by $\|\,\mathbf{W}_k\,\|_F \ll 1$, which would result in a misleading performance interpretation. From the simulation curves it is clearly seen that once the Newton algorithm approaches the vicinity of a minimum, it reveals a *superlinear convergence* and attains the minimum within a few steps.

5. Conclusion. The problem of joint approximate diagonalization of a set of positive definite matrices has become of great interest in blind signal separation applications. Most algorithms known for the joint diagonalization task impose some constraints on the diagonalization matrix $\mathbf{W}$, namely, that $\mathbf{W}$ needs to be (i) real, (ii) unitary, or (iii) square. We have derived a Newton algorithm for this problem which has none of these restrictions. We allow the diagonalization matrix $\mathbf{W}$ to be complex, nonunitary, and even rectangular.

The most general case where the diagonalization matrix $\mathbf{W}$ can be rectangular, instead of being square, is of particular interest in blind signal separation. This scenario occurs when access to more sensor signals than source signals is available. In this case the correlation matrices $\mathbf{R}_p$ are no longer positive definite; they only will be positive semidefinite. Algorithms that use the same cost function as given in (4), but constrain $\mathbf{W}$ to being square, require that $\mathbf{R}_p$ be positive definite; otherwise $\det\left(\mathbf{W}\,\mathbf{R}_p\,\mathbf{W}^H\right)$ becomes zero. Since our algorithm can also update a rectangular $M \times N$ matrix $\mathbf{W}$, where $M \leq N$, we impose a much weaker constraint, namely, that the product $\mathbf{W}\,\mathbf{R}_p\,\mathbf{W}^H$ needs to be positive definite and not $\mathbf{R}_p$. For a given set of $\{\mathbf{R}_p\}$ we can simply achieve this by reducing M, the number of rows of $\mathbf{W}$, until $\{\mathbf{W}\,\mathbf{R}_p\,\mathbf{W}^H\}$ has full rank for all p.

A major contribution of this paper is the derivation of the Hessian in closed form for every $\mathbf{W}$ and not only at the critical points. It turned out that the matrix form of the Taylor series expansion (12), as given by Manton in [13], has provided the foundation of this derivation. This form preserves the matrix structure of the underlying problem which allows a compact-form representation of the gradient and

Hessian through matrix and Kronecker products. Finally, we have shown that there exists a close similarity between the gradient and Hessian of two commonly used cost functions for the joint diagonalization problem.

Appendix A. Useful relations for deriving the gradient and Hessian of a matrix-valued cost function. The following equalities were very useful for the derivation of the gradient and Hessian. Some basic relations are

$$\| \mathbf{A} \|_F^2 = \mathrm{tr}\,(\mathbf{A}\mathbf{A}^H)\,, \tag{81}$$

$$\mathrm{tr}\,(\mathbf{A}\mathbf{B}) = \mathrm{tr}\,(\mathbf{B}\mathbf{A})\,, \tag{82}$$

$$\mathrm{tr}\left(\mathbf{A}^H\right) = \mathrm{tr}\,(\mathbf{A}^*) = (\mathrm{tr}\,(\mathbf{A}))^*\,. \tag{83}$$

Furthermore, we have some useful equalities with the vec(.) operation and Kronecker product [3] with $\mathbf{Z} \in \mathbb{C}^{M\times N}$:

$$\mathrm{vec}(\,\mathbf{Z}^T\,) = \mathbf{P}_{M\times N}\,\mathrm{vec}(\,\mathbf{Z}\,), \tag{84}$$

$$\mathrm{tr}(\mathbf{Z}^H\mathbf{A}) = \mathrm{vec}(\,\mathbf{Z}\,)^H\,\mathrm{vec}(\,\mathbf{A}\,), \tag{85}$$

$$\begin{aligned}\mathrm{tr}\,(\mathbf{Z}\mathbf{A}) &= \mathrm{vec}(\,\mathbf{Z}^T\,)^T\,\mathrm{vec}(\,\mathbf{A}\,) \\ &= \mathrm{vec}(\,\mathbf{Z}\,)^T\,\mathbf{P}_{M\times N}^T\,\mathrm{vec}(\,\mathbf{A}\,),\end{aligned} \tag{86, 87}$$

$$(\mathbf{A}\otimes\mathbf{B})^T = \mathbf{A}^T\otimes\mathbf{B}^T, \tag{88}$$

$$(\mathbf{A}\otimes\mathbf{B})^H = \mathbf{A}^H\otimes\mathbf{B}^H, \tag{89}$$

$$(\mathbf{A}\otimes\mathbf{B})^{-1} = \mathbf{A}^{-1}\otimes\mathbf{B}^{-1}, \tag{90}$$

$$(\,\mathbf{A}\,\mathbf{B}\otimes\mathbf{C}\,\mathbf{D}\,) = (\,\mathbf{A}\otimes\mathbf{C}\,)(\,\mathbf{B}\otimes\mathbf{D}\,), \tag{91}$$

$$\mathbf{A}^{P\times Q}\otimes\mathbf{B}^{R\times S} = \mathbf{P}_{P\times R}\,(\,\mathbf{B}\otimes\mathbf{A}\,)\,\mathbf{P}_{S\times Q}, \tag{92}$$

$$\mathrm{vec}(\,\mathbf{A}\,\mathbf{Z}\,\mathbf{B}\,) = (\,\mathbf{B}^T\otimes\mathbf{A}\,)\,\mathrm{vec}(\,\mathbf{Z}\,), \tag{93}$$

where the permutation matrix $\mathbf{P}_{M\times N}$ is uniquely defined with (84). Sometimes $\mathbf{P}_{M\times N}$ is called the *commutation matrix* [12]. Since $\mathbf{P}_{M\times N}$ is a permutation matrix, $\mathbf{P}_{M\times N} = \mathbf{P}_{N\times M}^T = \mathbf{P}_{N\times M}^{-1}$. For the special case where $M = N$, the commutation matrix is *involutory*, $\mathbf{P}_{M\times M}^2 = \mathbf{I}$, as $\mathbf{P}_{M\times M}^T = \mathbf{P}_{M\times M}$ is symmetric. See [3, 12] for a thorough list of properties of Kronecker products.

The following relations, where $\mathbf{Z}_1, \mathbf{Z}_2 \in \mathbb{C}^{M\times N}$ and the argument of tr (.) is a square matrix, have been proven to be very useful in the derivation of the Hessian:

$$\mathrm{tr}\,(\mathbf{Z}_1\,\mathbf{A}\,\mathbf{Z}_2\,\mathbf{B}) = \mathrm{vec}(\,\mathbf{Z}_1\,)^T\,\mathbf{P}_{M\times N}^T\,(\,\mathbf{B}^T\otimes\mathbf{A}\,)\,\mathrm{vec}(\,\mathbf{Z}_2\,), \tag{94}$$

$$\mathrm{tr}\left(\mathbf{Z}_1^H\mathbf{A}\,\mathbf{Z}_2\,\mathbf{B}\right) = \mathrm{vec}(\,\mathbf{Z}_1\,)^H\,(\,\mathbf{B}^T\otimes\mathbf{A}\,)\,\mathrm{vec}(\,\mathbf{Z}_2\,), \tag{95}$$

$$\mathrm{tr}\left(\mathbf{Z}_1^H\mathbf{A}\,\mathbf{Z}_2^H\,\mathbf{B}\right) = \mathrm{vec}(\,\mathbf{Z}_1\,)^H\,(\,\mathbf{B}^T\otimes\mathbf{A}\,)\,\mathbf{P}_{M\times N}\,\mathrm{vec}(\,\mathbf{Z}_2\,)^*. \tag{96}$$

Equation (95) can be derived with (85) and (93). Equations (94) and (96) can be derived with (85), (93), (87), and (83).

Appendix B. Armijo rule for matrix form. The Armijo rule for choosing a step size μ_k at the kth iteration is defined as $\mu_k = \mu_0\,\gamma^m$, where m is the first nonnegative integer that fulfills

$$\mathcal{J}(\mathbf{W}_k) - \mathcal{J}(\mathbf{W}_k + \mu_0\,\gamma^m\,\mathbf{S}_k) \geq -\eta\,\mu_0\,\gamma^m\,\mathfrak{Re}\{\langle\mathbf{S}_k, \mathbf{D}_{\mathbf{W}k}\rangle\}. \tag{97}$$

The search direction and the gradient of $\mathcal{J}$ at $\mathbf{W}_k$ are denoted as $\mathbf{S}_k$ and $\mathbf{D}_{\mathbf{W}k}$, respectively, and $\langle\mathbf{S}_k, \mathbf{D}_{\mathbf{W}k}\rangle \triangleq \mathrm{tr}\left(\mathbf{S}_k^H\mathbf{D}_{\mathbf{W}k}\right)$ defines an inner product between $\mathbf{S}_k$ and $\mathbf{D}_{\mathbf{W}k}$.

Acknowledgments. The author would like to thank Pascal Vontobel for helpful discussions and the anonymous reviewers for insightful comments and suggestions. They all helped to improve this paper.

REFERENCES

[1] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, *A blind source separation technique using second-order statistics*, IEEE Trans. Signal Process., 45 (1997), pp. 434–444.

[2] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[3] J. W. Brewer, *Kronecker products and matrix calculus in system theory*, IEEE Trans. Circuits and Systems, 25 (1978), pp. 772–781.

[4] J.-F. Cardoso and A. Souloumiac, *Blind beamforming for non Gaussian signals*, IEE Proceedings-F, 140 (1993), pp. 362–370.

[5] J.-F. Cardoso and A. Souloumiac, *Jacobi angles for simultaneous diagonalization*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 161–164.

[6] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley & Sons, New York, 1987.

[7] B. Flury, *Common principal components in k groups*, J. Amer. Statist. Assoc., 79 (1984), pp. 892–897.

[8] B. N. Flury and W. Gautschi, *An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 169–184.

[9] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1990.

[10] M. Joho and K. Rahbar, *Joint diagonalization of correlation matrices by using Newton methods with application to blind signal separation*, in IEEE Sensor Array and Multichannel Signal Processing Workshop, Rosslyn, VA, 2002, pp. 403–407.

[11] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed., Addison-Wesley, Reading, MA, 1989.

[12] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, 2nd ed., John Wiley & Sons, New York, 1999.

[13] J. H. Manton, *Optimisation algorithms exploiting unitary constraints*, IEEE Trans. Signal Process., 50 (2002), pp. 635–650.

[14] D. T. Pham, *Joint approximate diagonalization of positive definite Hermitian matrices*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1136–1152.

[15] E. Polak, *Optimization: Algorithms and Consistent Approximations*, Springer-Verlag, Berlin, 1997.

[16] R. Vollgraf and K. Obermayer, *Quadratic optimization for simultaneous matrix diagonalization*, IEEE Trans. Signal Process., 54 (2006), pp. 3270–3278.

[17] A. Yeredor, *Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation*, IEEE Trans. Signal Process., 50 (2002), pp. 1545–1553.

[18] A. Yeredor, *On using exact joint diagonalization for noniterative approximate joint diagonalization*, IEEE Signal Process. Lett., 12 (2005), pp. 645–648.

[19] A. Yeredor, A. Ziehe, and K. R. Müller, *Approximate joint diagonalization using a natural-gradient approach*, in Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation (ICA), Granada, Spain, 2004, pp. 89–96.

[20] A. Ziehe, P. Laskov, and K.-R. Müller, *A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation*, J. Mach. Learn. Res., (2004), pp. 777–800.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1219–1232

A JACOBI-TYPE METHOD FOR COMPUTING ORTHOGONAL TENSOR DECOMPOSITIONS*

CARLA D. MORAVITZ MARTIN† AND CHARLES F. VAN LOAN‡

Abstract. Suppose $\mathcal{A} = (a_{ijk}) \in \mathbb{R}^{n\times n\times n}$ is a three-way array or third-order tensor. Many of the powerful tools of linear algebra such as the singular value decomposition (SVD) do not, unfortunately, extend in a straightforward way to tensors of order three or higher. In the two-dimensional case, the SVD is particularly illuminating, since it reduces a matrix to diagonal form. Although it is not possible in general to diagonalize a tensor (i.e., $a_{ijk} = 0$ unless $i = j = k$), our goal is to "condense" a tensor in fewer nonzero entries using orthogonal transformations. We propose an algorithm for tensors of the form $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$ that is an extension of the Jacobi SVD algorithm for matrices. The resulting tensor decomposition reduces $\mathcal{A}$ to a form such that the quantity $\sum_{i=1}^{n} a_{iii}^2$ or $\sum_{i=1}^{n} a_{iii}$ is maximized.

Key words. multilinear algebra, tensor decomposition, singular value decomposition, multidimensional arrays

AMS subject classifications. 15A69, 65F30

DOI. 10.1137/060655924

1. Introduction.

The SVD of a matrix gives us important information about a matrix such as its rank, an orthonormal basis for the column or row space, and reduction to diagonal form. In applications, especially those involving multiway data analysis, information about the rank and reduction of tensors to have fewer nonzero entries are useful concepts to try to extend to higher dimensions.

Suppose $\mathcal{A} = (a_{ijk}) \in \mathbb{R}^{n\times n\times n}$ is a three-way array or third-order tensor. This paper is about computing a tensor decomposition of $\mathcal{A}$ such that $\mathcal{A}$ is written as a linear combination of rank-1 tensors of the form

$$w \otimes v \otimes u, \tag{1.1}$$

where $u, v, w \in \mathbb{R}^n$ and "$\otimes$" denotes the Kronecker product. The rank-1 tensor in (1.1) can also be denoted using the tensor outer product notation. In particular, (1.1) is equivalent to a vectorization of $u \circ v \circ w$.

The contribution of this paper is a higher-order generalization of the Jacobi SVD algorithm for matrices [14, p. 457] that works by solving small subproblems where $n = 2$. In the higher-order generalization, we find a tensor decomposition of the form

$$a = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \sigma_{ijk}(w_k \otimes v_j \otimes u_i), \tag{1.2}$$

where u_i, v_j, and w_k are the ith, jth, and kth columns of orthogonal matrices $U, V, W \in \mathbb{R}^{n\times n}$, respectively, and σ_{ijk} is the (i, j, k)th element of a tensor $\Sigma \in$

*Received by the editors March 31, 2006; accepted for publication (in revised form) by P. Comon December 17, 2007; published electronically September 25, 2008. This work was supported by NSF grant CCR-9901988.

http://www.siam.org/journals/simax/30-3/65592.html

†Department of Mathematics and Statistics, James Madison University, Harrisonburg, VA 22807 (carlam@math.jmu.edu).

‡Department of Computer Science, Cornell University, Ithaca, NY 14853-7510 (cv@cs.cornell.edu).

$\mathbb{R}^{n\times n\times n}$. The orthogonal matrices U, V, W are chosen to maximize either the sums of squares of the diagonals ($\sum_{i=1}^{n} \sigma_{iii}^2$) or the "trace" ($\sum_{i=1}^{n} \sigma_{iii}$).

Several Jacobi-like procedures have been implemented in the area of tensor decompositions already. In the case of symmetric tensors, there exist Jacobi algorithms to compute tensor decompositions of real or complex tensors [6, 7, 8, 12]. In particular, [12] shows that the solution to the Jacobi subproblem for symmetric tensors has an SVD solution of a particular symmetric matrix. We use a similar method when we show how to maximize the trace of a tensor. A Jacobi-type algorithm in [24] is used to simultaneously diagonalize positive definite Hermitian matrices using nonorthogonal transformations. This is relevant to tensors since [13] shows that a nonorthogonal tensor decomposition can be rewritten in terms of a simultaneous diagonalization problem for matrices. Furthermore, [13] proposes a Jacobi approach to simultaneous diagonalization (see also [4]) and shows that maximizing the sums of squares of the diagonals of a $2 \times 2 \times 2$ tensor using orthogonal transformations is equivalent to finding the roots of a polynomial of degree eight [13].

Tensor decompositions are used in many applications to help explain interactions among multiway data. These applications include chemometrics [3, 27], psychometrics [22], computer image and human motion recognition [29, 30], signal processing [25, 26], and many other areas using multiway data analyses [9]. Sometimes the tensor decompositions have a minimal number of terms in the linear combination, therefore "condensing" $\mathcal{A}$ into fewer nonzero entries so that interactions can be better explained.

A large amount of work has already been devoted to creating algorithms to compute orthogonal tensor decompositions. The most widely used algorithm is TUCKER3 originally proposed by Tucker [28]. Many improvements have been made to the algorithm since its original introduction, and the current version has been implemented in a MATLAB toolbox [1, 2]. A greedy algorithm to compute an orthogonal tensor decomposition has been proposed by [21], and [10] uses TUCKER3 to describe a higher-order generalization of the SVD for tensors. Several methods have also been developed to compute a "compressed" third-order orthogonal tensor decomposition, i.e., maximum variance of squares [16], maximum sums of squares of the diagonals of each face of a tensor [22], and maximum sums of squares of the diagonals of a third-order tensor [18].

A special case of TUCKER3 is the CANDECOMP-PARAFAC algorithm simultaneously proposed by [5] and [15]. Algorithms have also been developed to compute the nearest rank-1 tensor to a given tensor $\mathcal{A}$ (see [11, 20, 21, 32]). Furthermore, since the CANDECOMP-PARAFAC representation is equivalent to simultaneously diagonalizing a set of matrices, there are a number of recent algorithms related to simultaneous diagonalization (see [13] and the references therein).

Our presentation is organized as follows. First, we describe some matrix tools necessary to describe tensors and tensor decompositions in section 2. In section 3 we describe different ways to represent tensors. In sections 4 and 5 we describe the higher-order generalization of the Jacobi SVD algorithm—first for tensors $\mathcal{A} \in \mathbb{R}^{2\times 2\times 2}$ and then for general tensors $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$. We examine the algorithm cost in section 6, and describe a block version and an $\ell \times m \times n$ version in sections 7 and 8, respectively. Extending the algorithm to order-p tensors is discussed briefly in section 9. Finally, in section 10 we examine the performance of the algorithm.

2. Some properties of the Kronecker product. We review a few essential facts about the Kronecker product which are found in [31]. Computations that involve the Kronecker product require an understanding of the `vec` and `reshape` operators.

If $B \in \mathbb{R}^{m\times n}$, then $\texttt{vec}(B) \in \mathbb{R}^{mn}$ is the vector formed by "stacking" the columns of B.

The vec operator can be used to convert between matrix-vector and matrix-matrix products. For example, let $F \in \mathbb{R}^{m\times m}, G \in \mathbb{R}^{n\times n}, X \in \mathbb{R}^{n\times m}$. Then

$$Y = GXF^T \qquad \Longleftrightarrow \qquad \texttt{vec}(Y) = (F \otimes G)\texttt{vec}(X). \tag{2.1}$$

Another important property of vec involves outer products. For $x \in \mathbb{R}^m, y \in \mathbb{R}^n$, $\texttt{vec}(xy^T) = y \otimes x$.

The vec operator can be used in combination with the outer product to write certain matrix factorizations as vectors. For example, if $A \in \mathbb{R}^{m\times n}$ and $A = UBV^T$, where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$, then

$$A = \sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}u_iv_j^T \qquad \Longleftrightarrow \qquad \texttt{vec}(A) = \sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}(v_j \otimes u_i), \tag{2.2}$$

where u_i, v_j are the ith and jth columns of U and V, respectively.

We can also write $\texttt{vec}(A)$ as a matrix-vector product. If $a = \texttt{vec}(A)$ and $b = \texttt{vec}(B)$, then by (2.1)

$$a = (V \otimes U) \cdot b. \tag{2.3}$$

The reshape operator is a more general way of rearranging the entries in a matrix (it is also a MATLAB function). If $b \in \mathbb{R}^{mn}$, then $\texttt{reshape}(b, m, n)$ creates an $m \times n$ matrix from b. This operator is useful when converting between $B \otimes C$ and $C \otimes B$. If the vec *permutation matrix* $\Pi_{n,mn} \in \mathbb{R}^{mn\times mn}$ is defined by

$$\Pi_{n,mn} = \begin{bmatrix} I_{mn}(1:m:mn,:) \\ I_{mn}(2:m:mn,:) \\ \vdots \\ I_{mn}(m:m:mn,:) \end{bmatrix},$$

then it can be shown that (see [31])

$$\Pi_{n,mn}^T(B \otimes C)\Pi_{n,mn} = C \otimes B.$$

3. Tensor decompositions. We use the accepted notation where a third-order tensor is indexed by three indices and can be represented as a "cube" of data [19]. While the cube orientation is not unique, here we say that the kth index indicates the face of the cube. See Figure 1 for an illustration when $n = 2$.

An $n \times n \times n$ tensor $\mathcal{A}$ is also a three-way array where the kth face is represented using MATLAB notation as $A(:,:,k)$. The entries $\texttt{vec}(\mathcal{A})$ can also be rearranged to correspond to viewing the cube in different ways (see Figure 2 for an illustration when $n = 2$). Viewing the tensor cube in different orientations corresponds to a rearrangement of the elements of $\mathcal{A}$. This can be better described by a permutation of the elements of $\texttt{vec}(\mathcal{A})$ by a reshape operation. For example, if $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$ and $a = \texttt{vec}(\mathcal{A})$, then the different cuts can be represented in vector form as

$$\begin{aligned} a_1 &= \texttt{vec}(\texttt{reshape}(\Pi_{n,n^3}^T \cdot a, n, n^2)) && \text{(top-bottom)}, \\ a_2 &= \texttt{vec}(\texttt{reshape}(\Pi_{n^2,n^3}^T \cdot a, n, n^2)) && \text{(left-right)}, \\ a_3 &= \texttt{vec}(\texttt{reshape}(a, n, n^2)) && \text{(front-back)}, \end{aligned} \tag{3.1}$$

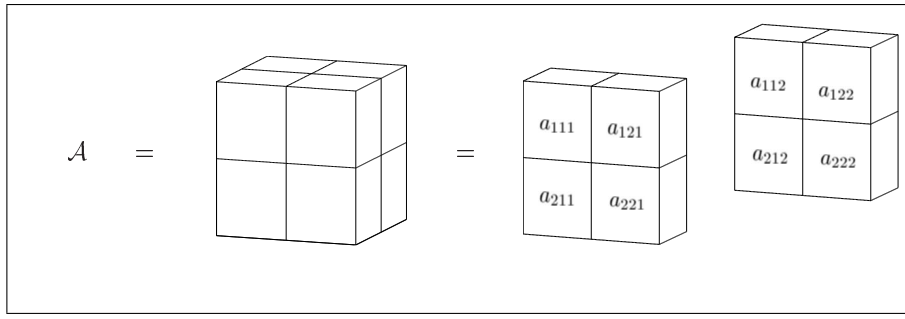


FIG. 1. *Illustration of a third-order tensor as a cube of data when* $n = 2$.

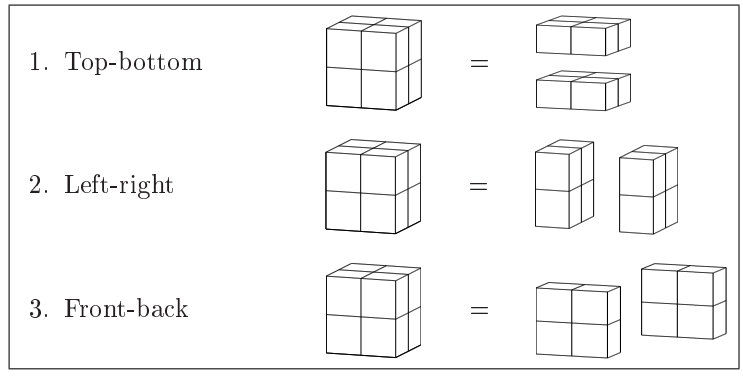


FIG. 2. *Three ways to cut a cube of data for a third-order tensor.*

where $a_3 = \texttt{vec}(\mathcal{A})$. Viewing the different faces of the cube side by side as matrices also corresponds to *unfolding matrices* [10] or *matricizations* [19].

The vec operator is used in the same way on tensors as on matrices. If $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$, then $\texttt{vec}(\mathcal{A}) \in \mathbb{R}^{n^3}$ is the vector formed by stacking the column vectors $\texttt{vec}(A(:,:,i))$ for $i = 1,\ldots,n$.

Hence, given $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$, the basic goal of this work is to find orthogonal matrices $U, V, W \in \mathbb{R}^{n\times n}$ such that

$$a \equiv \texttt{vec}(\mathcal{A}) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \sigma_{ijk}(w_k \otimes v_j \otimes u_i), \tag{3.2}$$

and either of the two quantities $\sum_{i=1}^{n} \sigma_{iii}^2$ or $\sum_{i=1}^{n} \sigma_{iii}$ are maximized. In (3.2), u_i, v_j, and w_k are the ith, jth, and kth columns of U, V, and W, respectively.

By (2.2) and (2.3), (3.2) can also be written as the matrix-vector product

$$a = (W \otimes V \otimes U) \cdot \sigma, \tag{3.3}$$

where $\sigma = \texttt{vec}(\Sigma)$ and $\Sigma = (\sigma_{ijk}) \in \mathbb{R}^{n\times n\times n}$. The representation in (3.3) will be used to describe our algorithm in the sections that follow.

One may ask whether it would be better to choose U, V, and W in (3.2) such that the tensor $\Sigma = (\sigma_{ijk})$ is "diagonal" (i.e., $\sigma_{ijk} = 0$ unless $i = j = k$). In this scenario, (3.2) reduces exactly to the matrix SVD in two dimensions. However, diagonality from orthogonal transformations is possible only for tensors of order three or higher

in special cases. In general it is not possible to find orthogonal matrices U, V, W such that the tensor Σ is diagonal [10].

4. `Jacobi-Compress` algorithm for $n \times n \times n$ tensors. We now describe `Jacobi-Compress`, the higher-order generalization of the Jacobi SVD algorithm in [14, p. 457]. The overall idea is to compute tensor decompositions of the form (3.2) of $2 \times 2 \times 2$ subtensors. Here, we explain the general procedure when $\mathcal{A} \in \mathbb{R}^{n \times n \times n}$. The next section details how to solve the $2 \times 2 \times 2$ subproblem.

In the spirit of the Jacobi SVD algorithm, the first step is to choose a (p, q) pair and form the corresponding $2 \times 2 \times 2$ subtensor $\tilde{A}$ given by

$$\tilde{A}(:,:,1) = \begin{bmatrix} a_{ppp} & a_{pqp} \\ a_{qpp} & a_{qqp} \end{bmatrix}, \qquad \tilde{A}(:,:,2) = \begin{bmatrix} a_{ppq} & a_{pqq} \\ a_{qpq} & a_{qqq} \end{bmatrix}.$$

Using the representation in (3.3), suppose we have orthogonal matrices $\tilde{U}, \tilde{V}, \tilde{W} \in \mathbb{R}^{2\times 2}$ such that

$$\begin{bmatrix} \sigma_{ppp} \\ \sigma_{qpp} \\ \sigma_{pqp} \\ \sigma_{qqp} \\ \sigma_{ppq} \\ \sigma_{qpq} \\ \sigma_{pqq} \\ \sigma_{qqq} \end{bmatrix} = (\tilde{W}^T \otimes \tilde{V}^T \otimes \tilde{U}^T) \begin{bmatrix} a_{ppp} \\ a_{qpp} \\ a_{pqp} \\ a_{qqp} \\ a_{ppq} \\ a_{qpq} \\ a_{pqq} \\ a_{qqq} \end{bmatrix}, \tag{4.1}$$

and either $(\sigma_{ppp}^2 + \sigma_{qqq}^2)$ or $(\sigma_{ppp} + \sigma_{qqq})$ is maximized. To update $a = \texttt{vec}(\mathcal{A}) \in \mathbb{R}^{n^3}$, set U to be the $n \times n$ identity matrix, except that $I(p:q, p:q) = \tilde{U}$ and analogously for V and W. Then perform the update

$$\sigma \leftarrow (W^T \otimes V^T \otimes U^T) \cdot a, \tag{4.2}$$

where $a = [\texttt{vec}(\tilde{A}(:,:,1)); \texttt{vec}(\tilde{A}(:,:,2))]$. In practice, significant savings are achieved by observing that the only elements of a that change are those with a p or q in the index; i.e., the pth and qth front-back faces, the pth and qth side faces, and the pth and qth top-bottom faces. This avoids the actual computation of the above Kronecker product.

Rather than just repeating *sweeps* through all the possible (p, q) pairs as in the Jacobi SVD algorithm, we alternate the view or orientation of the cube (Figure 2) at each sweep. Therefore, one *iteration* of `Jacobi-Compress` includes three sweeps: one sweep for each orientation. Changing the orientation simply involves a `reshape` operation defined in (3.1). `Jacobi-Compress` is complete when an iteration does not significantly change (within some specified tolerance) the sums of squares of the diagonals or the trace of the tensor.

5. The $2 \times 2 \times 2$ subproblem. Suppose $\mathcal{A} \in \mathbb{R}^{2\times 2\times 2}$ and $a = \texttt{vec}(\mathcal{A})$. We now show how to solve (4.1). That is, the goal is to find 2×2 orthogonal matrices $\tilde{U}$, $\tilde{V}$, and $\tilde{W}$ such that

$$\sigma = (\tilde{W}^T \otimes \tilde{V}^T \otimes \tilde{U}^T)a$$

has maximum $(\sigma_{111}^2 + \sigma_{222}^2)$ or $(\sigma_{111} + \sigma_{222})$.

The idea behind solving the subproblem is an iterative approach that differs based on whether we are maximizing the trace or sums of squares. However, both approaches involve taking an SVD of a specific 2×2 matrix.

5.1. Maximizing the trace. Maximizing the trace involves holding one variable constant while varying the others. For example, suppose we have performed the following three steps:

$$\begin{aligned} \sigma_1 &\leftarrow (I \otimes \tilde{V}_1^T \otimes \tilde{U}_1^T)a, \\ \sigma_2 &\leftarrow (\tilde{W}_1^T \otimes I \otimes \tilde{U}_2^T)\sigma_1, \\ \sigma_3 &\leftarrow (\tilde{W}_2^T \otimes \tilde{V}_2^T \otimes I)\sigma_2. \end{aligned} \tag{5.1}$$

Then a tensor decomposition of a has been computed since

$$\begin{aligned} \sigma_3 &= (\tilde{W}_2^T \otimes \tilde{V}_2^T \otimes I)(\tilde{W}_1^T \otimes I \otimes \tilde{U}_2^T)(I \otimes \tilde{V}_1^T \otimes \tilde{U}_1^T)a \\ &= (\tilde{W}_2^T\tilde{W}_1^T \otimes \tilde{V}_2^T\tilde{V}_1^T \otimes \tilde{U}_2^T\tilde{U}_1^T)a \\ &= (\tilde{W}^T \otimes \tilde{V}^T \otimes \tilde{U}^T)a, \end{aligned}$$

where $\tilde{U} = \tilde{U}_1\tilde{U}_2$, $\tilde{V} = \tilde{V}_1\tilde{V}_2$, and $\tilde{W} = \tilde{W}_1\tilde{W}_2$.

The steps in (5.1) illustrate the central idea behind our algorithm. After each step, the trace of σ is maximized. The updates (5.1) are all algorithmically equivalent since

$$\Pi_{4,8}^T(\tilde{W}^T \otimes I \otimes \tilde{U}^T)\Pi_{4,8} = (I \otimes \tilde{U}^T \otimes \tilde{W}^T),$$

$$\Pi_{2,8}^T(\tilde{W}^T \otimes \tilde{V}^T \otimes I)\Pi_{2,8} = (I \otimes \tilde{W}^T \otimes \tilde{V}^T).$$

Therefore it suffices to describe how to find $\tilde{U}, \tilde{V}$ such that

$$\sigma = (I \otimes \tilde{V}^T \otimes \tilde{U}^T)a. \tag{5.2}$$

A remark about (5.2) is necessary. Equation (5.2) can be rewritten in terms of matrices as

$$\begin{aligned} \begin{bmatrix} \sigma_{111} & \sigma_{121} \\ \sigma_{211} & \sigma_{221} \end{bmatrix} &= \tilde{U}^T \begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix} \tilde{V}, \\ \begin{bmatrix} \sigma_{112} & \sigma_{122} \\ \sigma_{212} & \sigma_{222} \end{bmatrix} &= \tilde{U}^T \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix} \tilde{V}. \end{aligned} \tag{5.3}$$

However, we emphasize that solving (5.3) is not a joint SVD problem. The joint SVD problem [17, 23] uses a least squares approach to find matrices $\tilde{U}, \tilde{V}$ that diagonalize the left two matrices above. A Jacobi-like algorithm was proposed in [23] that reduces the problem to finding joint SVDs of 2×2 matrices by maximizing the sums of squares of the diagonals. In [23] it is shown that the joint SVD problem with 2×2 matrices has an explicit solution, namely, that maximizing the sums of squares is equivalent to maximizing the trace. In our case, we are maximizing the sums of squares or trace of the *tensor* diagonals. The two problems are not equivalent in our case.

The solution depends on whether $\tilde{U}$ and $\tilde{V}$ are both rotation or reflection matrices or whether one is a rotation matrix and one is a reflection matrix.

First, suppose $\tilde{U}_1$ and $\tilde{V}_1$ are the rotation matrices

$$\tilde{U}_1 = \begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix}, \qquad \tilde{V}_1 = \begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix}. \tag{5.4}$$

Then

$$\begin{aligned}\operatorname{tr}(\sigma) &= c_1c_2(a_{111}+a_{222}) + s_1c_2(-a_{211}+a_{122}) \\ &\quad + c_1s_2(-a_{121}+a_{212}) + s_1s_2(a_{221}+a_{112}).\end{aligned} \tag{5.5}$$

If

$$B_1 = \begin{bmatrix} a_{111}+a_{222} & a_{121}-a_{212} \\ a_{211}-a_{122} & a_{221}+a_{112} \end{bmatrix}, \tag{5.6}$$

then the $(1,1)$-entry of $\tilde{U}_1^T B_1 \tilde{V}_1$ is exactly (5.5). Therefore, maximizing (5.5) is equivalent to finding the SVD of B_1, since the largest singular value is the largest $(1,1)$-entry possible. The same result can be derived if $\tilde{U}_1$ and $\tilde{V}_1$ are both reflection matrices. In this scenario, we must ensure that the SVD consists of either both rotation matrices or both reflection matrices. Using an SVD solution to the trace problem is similar to a method used in [23] to compute the joint SVD.

Now, suppose that $\tilde{U}_2$ is a rotation matrix and $\tilde{V}_2$ is a reflection matrix:

$$\tilde{U}_2 = \begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix}, \qquad \tilde{V}_2 = \begin{bmatrix} c_2 & s_2 \\ s_2 & -c_2 \end{bmatrix}. \tag{5.7}$$

Then

$$\begin{aligned}\operatorname{tr}(\sigma) &= c_1c_2(a_{111}-a_{222}) + s_1c_2(a_{211}+a_{122}) \\ &\quad + c_1s_2(-a_{121}-a_{212}) + s_1s_2(a_{112}-a_{221}).\end{aligned} \tag{5.8}$$

If

$$B_2 = \begin{bmatrix} a_{111}-a_{222} & a_{121}+a_{212} \\ a_{211}+a_{122} & a_{221}-a_{112} \end{bmatrix}, \tag{5.9}$$

then the $(1,1)$-entry of $\tilde{U}_2^T B_2^T \tilde{V}_2$ is exactly (5.8), and therefore maximizing (5.8) is equivalent to taking the SVD of B_2. The same result holds if $\tilde{U}_2$ is a reflection matrix and $\tilde{V}_2$ is a rotation matrix.

Comparing the (1,1)-entries of B_1 and B_2 determines how $\tilde{U}$ and $\tilde{V}$ are chosen. For example, if

$$a_{111}+a_{222} > a_{111}-a_{222},$$

then we use B_1 and the SVD should involve either both rotation matrices or both reflection matrices. On the other hand, if

$$a_{111}+a_{222} < a_{111}-a_{222},$$

then we compute the SVD of B_2 and the result should involve one rotation matrix and one reflection matrix.

5.2. Maximizing the sums of squares. Setting up the problem to maximizing the sums of squares in a similar way to section 5.1 involves solving a nonlinear optimization problem that does not have a straightforward explicit solution. Here, we instead hold two variables constant and vary the third, which results in an explicit solution involving the SVD. The basic problem requires solving

$$\begin{aligned}\sigma_1 &\leftarrow (I \otimes I \otimes \tilde{U}^T)a, \\ \sigma_2 &\leftarrow (I \otimes \tilde{V}^T \otimes I)\sigma_1, \\ \sigma_3 &\leftarrow (\tilde{W}^T \otimes I \otimes I)\sigma_2.\end{aligned} \tag{5.10}$$

Similar to section 5.1, each of the steps above are equivalent since they involve permuting the tensor elements at each step. We therefore describe how to find $\tilde{U}$ such that

$$\sigma = (I \otimes I \otimes \tilde{U})\, a = \begin{bmatrix} \tilde{U} & & & \\ & \tilde{U} & & \\ & & \tilde{U} & \\ & & & \tilde{U} \end{bmatrix} \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix}. \tag{5.11}$$

From (5.11), it suffices to find orthogonal $\tilde{U}$ that maximizes the sums of squares of the diagonals of the matrix $\tilde{U}A$, where

$$A = \begin{bmatrix} a_{111} & a_{122} \\ a_{211} & a_{222} \end{bmatrix}. \tag{5.12}$$

Suppose the SVD of A is given by

$$A = U \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix},$$

where (c, s) is a cosine/sine pair. Indeed, if we have a 2×2 orthogonal matrix Z such that the sums of squares of the diagonals of

$$Z \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \tag{5.13}$$

are maximized, then we set $\tilde{U} = ZU^T$. We now explain how to obtain Z.

Without loss of generality, assume that Z is the rotation matrix

$$Z = \begin{bmatrix} \tilde{c} & \tilde{s} \\ -\tilde{s} & \tilde{c} \end{bmatrix}.$$

From (5.13) we are trying to find the pair $(\tilde{c}, \tilde{s})$ that maximizes

$$(\sigma_1 \tilde{c} c - \sigma_2 \tilde{s} s)^2 + (-\sigma_1 \tilde{s} s + \sigma_2 \tilde{c} c)^2. \tag{5.14}$$

After some simplification, (5.14) is equivalent to

$$(\sigma_1^2 + \sigma_2^2) - \left\| M \begin{bmatrix} \tilde{c} \\ \tilde{s} \end{bmatrix} \right\|_2^2, \tag{5.15}$$

where

$$M = \begin{bmatrix} \sigma_2 s & \sigma_1 c \\ \sigma_1 s & \sigma_2 c \end{bmatrix}. \tag{5.16}$$

Since

$$\sigma_1^2 + \sigma_2^2 = \|M\|_F^2 = \sigma_{max}(M)^2 + \sigma_{min}(M)^2, \tag{5.17}$$

where $\sigma_{max}(M)$ and $\sigma_{min}(M)$ are the largest and smallest singular values of M, respectively, we choose $[\tilde{c}, \tilde{s}]^T$ to be the right singular vector of M associated to the smallest singular value. This also means that the maximum sums of squares of the diagonals of A are equal to $\sigma_{max}(M)^2$. The same result is obtained if Z is chosen to be a reflection matrix.

$[\tilde{U}, \tilde{V}, \tilde{W}, \sigma] = \textbf{Solve2-by-2-by-2}(\mathcal{A})$

$a_0 = \texttt{vec}(\mathcal{A})$
for $j = 1, 2, 3$ (each dimension) **do**

% Solves $\sigma_j = (I \otimes Y_j^T \otimes X_j^T)\sigma_{j-1}$
$B \leftarrow \texttt{reshape}(\sigma_{j-1}, 2, 4)$
$\Sigma_1 \leftarrow B(1:2, 1:2)$
$\Sigma_2 \leftarrow B(1:2, 3:4)$

if sums of squares are maximized **then**

$A \leftarrow [\Sigma_1(:,1)\ \ \Sigma_2(:,2)]$
$[U, S, V] = \textbf{svd}(A)$

$$M \leftarrow \begin{bmatrix} \sigma_2 s & \sigma_1 c \\ \sigma_1 s & \sigma_2 c \end{bmatrix} \quad \text{(See (5.16))}$$

$[\bar{U}, \bar{S}, \bar{V}] = \textbf{svd}(M)$

$$Z \leftarrow \begin{bmatrix} \bar{v}_{12} & \bar{v}_{22} \\ -\bar{v}_{22} & \bar{v}_{12} \end{bmatrix}$$

$$X_j \leftarrow ZU^T; \quad Y_j \leftarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};$$

$\sigma_j \leftarrow \texttt{vec}([X_j\Sigma_1 \mid X_j\Sigma_2])$

else if trace is maximized **then**
if $a_{111} + a_{222} \geq a_{111} - a_{222}$ **then**

$$B \leftarrow \begin{bmatrix} a_{111} + a_{222} & a_{121} - a_{212} \\ a_{211} - a_{122} & a_{221} + a_{112} \end{bmatrix}$$

else

$$B \leftarrow \begin{bmatrix} a_{111} - a_{222} & a_{121} + a_{212} \\ a_{211} + a_{122} & a_{221} - a_{112} \end{bmatrix}$$

end if

$[X_j, S, Y_j] = \textbf{svd}(B)$

$\sigma_j \leftarrow \texttt{vec}\left([\ X_j^T\Sigma_1 Y_j \mid X_j\Sigma_2 Y_j\ ]\right)$

end if

end for
$\sigma \leftarrow \sigma_3; \quad \tilde{W} \leftarrow X_2 Y_3; \quad \tilde{V} \leftarrow Y_1 X_3; \quad \tilde{U} \leftarrow X_1 Y_2;$

FIG. 3. *Algorithm to solve the* $2 \times 2 \times 2$ *subproblem.*

6. Algorithm cost. Figures 3 and 4 contain `Jacobi-Compress` and its corresponding $2 \times 2 \times 2$ subproblem in pseudo-MATLAB. Actual MATLAB code can be found at [33].

To assess the amount of work, note that each *iteration* of `Jacobi-Compress` includes three *sweeps* through the cube, and each sweep involves $n(n-1)/2$ (p, q)-

$[U, V, W, \sigma] =$ **Jacobi-Compress**$(\mathcal{A})$

$\sigma \leftarrow \texttt{vec}(\mathcal{A})$
$U \leftarrow I_n; \quad V \leftarrow I_n; \quad W \leftarrow I_n;$
$X \leftarrow I_n; \quad Y \leftarrow I_n; \quad Z \leftarrow I_n;$
repeat
  $\sigma_0 \leftarrow \sigma$
  for $s = 1, 2, 3$ (each view of tensor cube) **do**
    % One sweep of the cube
    for $p = 1 : n-1$ **do**
      for $q = p+1 : n$ **do**

        Set $\Sigma \in \mathbb{R}^{2\times 2\times 2}$ to be the (p, q)-subtensor of σ_{s-1}

        $[\tilde{U}, \tilde{V}, \tilde{W}, \tilde{\sigma}] \leftarrow$ **Solve2-by-2-by-2**(Σ)

        % update other entries
        $W \leftarrow I_n; \quad V \leftarrow I_n; \quad U \leftarrow I_n$

        Update (p, q)-entries of σ_{s-1} with $\tilde{\sigma}$

        $W(p:q, p:q) \leftarrow W$
        $V(p:q, p:q) \leftarrow V$
        $U(p:q, p:q) \leftarrow U$
        $\sigma_s \leftarrow (W^T \otimes V^T \otimes U^T)\, \sigma_{s-1}$

        $X(:, [p \;\; q]) \leftarrow X(:, [p \;\; q]) \cdot U$
        $Y(:, [p \;\; q]) \leftarrow Y(:, [p \;\; q]) \cdot V;$
        $Z(:, [p \;\; q]) \leftarrow Z(:, [p \;\; q]) \cdot W;$
      end for
    end for
    $U_s \leftarrow X; \quad V_s \leftarrow Y; \quad W_s \leftarrow Z;$
  end for
  $\sigma \leftarrow \hat{\sigma}_3$
  $U \leftarrow U \cdot X_1 Z_2 Y_3$
  $V \leftarrow V \cdot Y_1 X_2 Z_3$
  $W \leftarrow W \cdot Z_1 Y_2 X_3$
until convergence

FIG. 4. *Jacobi-Compress for $n \times n \times n$ tensors.*

pairs. Solving the $2\times2\times2$ subproblem is constant work in both the case of maximizing the sums of squares and the trace. It is important to note that the update (4.2) can be computed in linear time by only updating those elements that are affected. Therefore, one iteration of `Jacobi-Compress` is $\mathcal{O}(n^3)$.

7. Block version. `Jacobi-Compress` can be converted to a block algorithm by representing an $n\times n\times n$ tensor as an $N\times N\times N$ block tensor with block size $r\times r\times r$, where $n = Nr$. For example, a $6\times6\times6$ tensor $\mathcal{A}$ can be regarded as a $3\times3\times3$ block

tensor with $2 \times 2 \times 2$ entries. The block version chooses a (p, q) pair, a $2r \times 2r \times 2r$ tensor, to be solved by `Jacobi-Compress`. Therefore, the only difference between the block version and standard `Jacobi-Compress` is the order in which the subproblems are solved.

8. Extension to $\ell \times m \times n$ tensors. Thus far, we have considered only three-way tensors for which each dimension is equal. Many applications involve data with unequal dimensions. By padding the data with zeros, `Jacobi-Compress` can be used for $\ell \times m \times n$ tensors.

Specifically, if $\tilde{p} = \max(\ell, m, n)$, then we pad the tensor with zeros that results in a $\tilde{p} \times \tilde{p} \times \tilde{p}$ tensor. Similar to the Jacobi SVD algorithm, unwanted fill does not occur. Note that computational shortcuts are taken so as to not actually store and perform calculations with zeros. In particular, if, say, $\ell >> m, n$, then many of the subproblems will contain all zeros. Therefore a sweep may contain considerably fewer subproblems than $\tilde{p}(\tilde{p} - 1)/2$.

9. Extension to order-p tensors. `Jacobi-Compress` can be extended to p-way tensors, but the results are not very practicable as p gets large. Maximizing the sums of squares of the diagonals is a direct extension of the $p = 3$ case in section 5.2; i.e., it involves taking SVDs of p 2×2 matrices. Unfortunately, extending the trace solution of section 5.1 does not involve an explicit solution to the subproblem but instead requires the help of a nonlinear solver to solve the optimization problem. More work is needed in this area.

10. Algorithm performance. In this section we describe the performance and numerical convergence typically seen in practice. In randomly generated examples, `Jacobi-Compress` typically converges in three iterations or less, i.e., the sums of squares of the diagonals or trace do not improve (up to a specified tolerance) after three iterations of the algorithm. We also note that in cases where a tensor is orthogonally diagonalizable, our algorithm finds that optimal form. The next example shows the compression of the algorithm.

Example 10.1. Let $\mathcal{A} \in \mathbb{R}^{3\times3\times3}$ be given by (in order of first face, second face, third face)

$$\mathcal{A} = \left[\begin{array}{ccc|ccc|ccc} 8 & 8 & 3 & 10 & 8 & 10 & 9 & 3 & 4 \\ 10 & 5 & 7 & 8 & 3 & 7 & 7 & 7 & 6 \\ 10 & 5 & 4 & 5 & 5 & 3 & 2 & 7 & 5 \end{array}\right].$$

Then `Jacobi-Compress` produces

$$\Sigma = \left[\begin{array}{ccc|ccc|ccc} \mathbf{.15} & -1.1 & 1.4 & 2.7 & .01 & -1.8 & -1.8 & -1.5 & .05 \\ 3.2 & .01 & .63 & 0 & \mathbf{33.4} & .03 & -.66 & .16 & -1.6 \\ -2.2 & 4.8 & .04 & -.17 & .30 & -.83 & -.03 & -.17 & \mathbf{-6.2} \end{array}\right].$$

One way to measure the compression of the algorithm is to look at the the "percent of norm" of the elements in the diagonals. For example, for an $n \times n \times n$ tensor, we can compute

$$\gamma = \frac{\sum_{i=1}^{n} \sigma_{iii}^2}{\sum_{i,j,k=1}^{n} \sigma_{ijk}^2},$$

which measures "how much" of the norm is contained in the diagonals of the tensor. The closer γ is to one, the better the compression. If the tensor is diagonalizable with

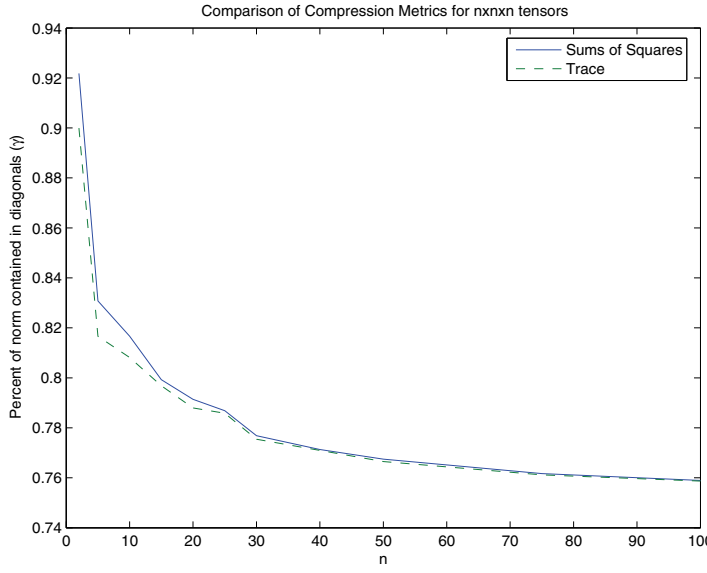


FIG. 5. *Percent of total norm obtained in the diagonal entries for randomly generated starting vectors using different metrics for compression for $n \times n \times n$ tensors.*

orthogonal transformations, γ should be equal to one. Figure 5 shows the average value of γ for different n after running `Jacobi-Compress` for $p = 3$ and the direct extension when $p = 4$. For comparison, γ tends to zero as n increases for random vectors. The clear trend from Figure 5 is that compression decreases as n increases. The inverse relationship between the amount of compression and n is expected since we have relatively fewer elements with which to explain n^3 or n^4 entries of the tensor. More importantly, however, is that the compression is quite high and tends to be around 75 percent compared to before running `Jacobi-Compress`, which is the result we focus attention to in Figure 5.

11. Conclusion. In this paper an algorithm based on the Jacobi SVD algorithm for matrices is given to compute an orthogonal tensor decomposition. The resulting tensor decomposition maximizes the sums of the squares of the "diagonals" or the trace of the tensor. The idea of the algorithm is to compute tensor decompositions of $2 \times 2 \times 2$ subtensors using an alternating least squares approach. In each case, the subproblem has an explicit solution involving the SVD. The general p-way case is still being explored.

REFERENCES

[1] C. A. ANDERSSON AND R. BRO, *Improving the speed of multi-way algorithms: Part* I. *Tucker*3, Chemom. Intell. Lab. Syst., 42 (1998), pp. 93–103.

[2] C. A. ANDERSSON AND R. BRO, *The N-way toolbox for MATLAB*, Chemom. Intell. Lab. Syst., 52 (2000), pp. 1–4.

[3] R. BRO, *PARAFAC. Tutorial and applications*, Chemom. Intell. Lab. Syst., 38 (1997), pp. 149–171.

[4] J. F. CARDOSO AND A. SOULOUMIAC, *Blind beam-forming for non-Gaussian signals*, IEEE Proceedings, Part F, 140 (1993), pp. 362–370.

[5] J. D. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[6] P. Comon, *Independent component analysis, a new concept?*, Signal Process., 36 (1994), pp. 287–314.

[7] P. Comon, *Tensor diagonalization, a useful tool in signal processing*, in Proceedings of the 10th IFAC Symposium on System Identification, Vol. 1, M. Blanke and T. Soderstrom, eds., IFAC–SYSID, Copenhagen, 1994, pp. 77–82.

[8] P. Comon, *Canonical Tensor Decompositions*, I3S report, RR-2004-17, CNRS-Laboratoire I3S, Université Nice-Sophia Antipolis, France, 2004.

[9] R. Coppi and S. Bolasco, eds., *Multiway Data Analysis*, Elsevier, Amsterdam, 1989.

[10] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[11] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[12] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Independent component analysis and (simultaneous) third-order tensor diagonalization*, IEEE Trans. Signal Process, 49 (2001), pp. 2262–2271.

[13] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 295–327.

[14] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[15] R. A. Harshman, *Foundations of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[16] R. Henrion and C. A. Andersson, *A new criteria for simple-structure transformations of core arrays in N-way principal component analysis*, Chemom. Intell. Lab. Syst., 47 (1999), pp. 190–204.

[17] G. Hori, *A general framework for SVD flows and joint SVD flows*, in Proceedings of the IEEE International Conference of Acoustics, Speech, and Signal Processing, Hong Kong, 2003, pp. 693–696.

[18] H. A. L. Kiers, *Tuckals core rotations and constrained Tuckals modelling*, Statist. Appl., 4 (1992), pp. 661–667.

[19] H. A. L. Kiers, *Towards a standardized notation and terminology in multiway analysis*, J. Chemom., 14 (2000), pp. 105–122.

[20] E. Kofidis and P. A. Regalia, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 863–884.

[21] T. G. Kolda, *Orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 243–255.

[22] P. M. Kroonenberg, *Three-mode Principal Component Analysis: Theory and Applications*, DSWO Press, Leiden, The Netherlands, 1983.

[23] B. Pesquet-Popescu, J.-C. Pesquet, and A. P. Petropulu, *Joint singular value decomposition—a new tool for separable representation of images*, in Proceedings of the IEEE International Conference of Image Processing, Thessaloniki, Greece, 2001, pp. 569–572.

[24] D. T. Pham, *Joint approximate diagonalization of positive definite Hermitian matrices*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1136–1152.

[25] N. Sidiropoulos, G. Giannakis, and R. Bro, *Blind PARAFAC receivers for DS-CDMA systems*, IEEE Trans. Signal Process., 48 (2000), pp. 810–823.

[26] N. Sidiropoulos, R. Bro, and G. Giannakis, *Parallel factor analysis in sensor array processing*, IEEE Trans. Signal Process., 48 (2000), pp. 2377–2388.

[27] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis: Applications in the Chemical Sciences*, John Wiley & Sons, Chichester, England, 2004.

[28] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[29] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear image analysis for face recognition*, in Proceedings of the International Conference on Pattern Recognition (ICPR), Quebec City, Canada, 2002, pp. 511–514.

[30] M. A. O. Vasilescu, *Human motion signatures: Analysis, synthesis, recognition*, in Proceedings of the International Conference on Pattern Recognition (ICPR), Quebec City, Canada, 2002, pp. 456–460.

[31] C. F. Van Loan, *The ubiquitous Kronecker product*, J. Comput. Appl. Math., 123 (2000), pp. 85–100.
[32] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.
[33] Matlab, http://www.math.jmu.edu/∼carlam.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1233–1253

TENSOR-PRODUCT APPROXIMATION TO MULTIDIMENSIONAL INTEGRAL OPERATORS AND GREEN'S FUNCTIONS*

WOLFGANG HACKBUSCH† AND BORIS N. KHOROMSKIJ†

Abstract. The Kronecker tensor-product approximation combined with the $\mathcal{H}$-matrix techniques provides an efficient tool to represent integral operators as well as a discrete elliptic operator inverse $A^{-1} \in \mathbb{R}^{N\times N}$ in $\mathbb{R}^d$ (the discrete Green's function) with a high spatial dimension d. In the present paper we give a survey on modern methods of the structured tensor-product approximation to multidimensional integral operators and Green's functions and present some new results on the existence of low tensor-rank decompositions to a class of function-related operators. The memory space of the considered data-sparse representations is estimated by $\mathcal{O}(dn\log^q n)$ with q independent of d, retaining the approximation accuracy of order $O(n^{-\delta})$, where $n = N^{1/d}$ is the dimension of the discrete problem in *one* space direction. In particular, we apply the results to the Newton, Yukawa, and Helmholtz kernels $\frac{1}{|x-y|}$, $\frac{e^{-\lambda|x-y|}}{|x-y|}$, and $\frac{\cos(\lambda|x-y|)}{|x-y|}$, respectively, with $x, y \in \mathbb{R}^d$.

Key words. hierarchical matrices, Kronecker tensor-product, Sinc approximation, integral operators, high dimensions

AMS subject classifications. 65F50, 65F30, 46B28, 47A80

DOI. 10.1137/060657017

1. Introduction.

In a wide range of applications one deals with quantities described by higher-order tensors, which are the higher-order analogues of vectors and matrices. Examples are multidimensional integral equations, elliptic, parabolic, and hyperbolic boundary value problems posed in $\mathbb{R}^d$, $d \geq 2$, Lyapunov and Riccati matrix equations, computation of spectral projection operators associated with the density matrix ansatz for solving the Hartree–Fock equation, as well as collision integrals of the deterministic Boltzmann equation. A naive numerical implementation of the corresponding multilinear algebra in the dimension $N = n^d$ with large d suffers from the so-called curse of dimensionality because of the exponential scaling in d. This phenomenon can be relaxed by invoking various data-sparse tensor-product formats to represent the fully populated higher-order tensors.

As a result of developments over more than three decades, we now have several well-established concepts of structured representation to higher-order tensors, which are based either on the so-called Tucker model [27] or on the CANDECOMP/PARAFAC (CP) decomposition [3, 21] (see section 2.1 for definitions). There are numerous successful applications of the Tucker and CP models in higher-order statistics, independent component analysis, chemometrics, telecommunications, signal processing, data mining, mathematical biology, complexity theory, and many other fields.

In the present paper we give a survey on modern methods of the structured tensor-product approximation to multidimensional integral operators and Green's functions and present some new results on the existence of low tensor-rank decompositions to a class of function-related operators. In particular, we focus on the construction of exponentially convergent in the tensor-rank decompositions. The asymptotic complexity of considered data-sparse representations is estimated by $\mathcal{O}(dn\log^q n)$ with q

*Received by the editors April 11, 2006; accepted for publication (in revised form) by N. Mastronardi December 20, 2006; published electronically September 25, 2008.
http://www.siam.org/journals/simax/30-3/65701.html

†Max-Planck-Institute for Mathematics in the Sciences, Inselstr. 22-26, D-04103 Leipzig, Germany (wh@mis.mpg.de, bokh@mis.mpg.de).

independent of d, where $n = N^{1/d}$ is the dimension of the discrete problem in *one* space direction. In particular, we apply the results to the Newton, the Yukawa, and the Helmholtz potentials $\frac{1}{|x-y|}$, $\frac{e^{-\lambda|x-y|}}{|x-y|}$, and $\frac{\cos(\lambda|x-y|)}{|x-y|}$, respectively, with $x, y \in \mathbb{R}^d$.

The class of *hierarchical ($\mathcal{H}$) matrices* allows an approximate matrix arithmetic with almost linear complexity [14, 15, 16, 12]. An $\mathcal{H}$-matrix approximation of the class of operator-valued functions of elliptic operators was developed in [8, 9, 10, 13]. For multidimensional problems, even approximations with linear complexity $\mathcal{O}(n^d)$ are not satisfactory. To avoid an exponential scaling in d, one can try to represent the corresponding data (matrices and vectors) in a *tensor-product form* (cf. [1, 19, 28]) to reach the complexity $\mathcal{O}(dn\log^q n)$ with $q \geq 1$ independent of d (see section 2.2). For this purpose the $\mathcal{H}$-matrix approach can be combined with the tensor-product approximations.

The *hierarchical Kronecker tensor-product* (HKT) representation of an integral operator $\mathbb{G} : L^2(\Omega) \to L^2(\Omega)$,

$$(\mathbb{G}u)(x) := \int_\Omega g(x,y)u(y)\mathrm{d}y, \quad x, y \in \Omega := [0,1]^d \in \mathbb{R}^d,$$

is based on a separable approximation to the explicitly given kernel function $g(x,y)$ accomplished with an $\mathcal{H}$-matrix representation of the canonical factors (cf. [19, 17, 18]).

In [11, 18], the $\mathcal{H}$-matrix techniques combined with the Kronecker tensor-product approximation (cf. [19, 28]) were applied to represent the inverse of a discrete elliptic operator $\mathbb{L}^{-1} : H^{-1}(\Omega) \to H_0^1(\Omega)$ in a hypercube $\Omega := (0,1)^d \in \mathbb{R}^d$,

$$\mathbb{L} = -\sum_{j=1}^d \frac{\partial}{\partial x_j} a_j(x_j) \frac{\partial}{\partial x_j} + \sum_{j=1}^d \left[b_j(x_j)\frac{\partial}{\partial x_j} + c_j(x_j) \right], \quad x = (x_1, \ldots, x_d) \in \Omega.$$

The approach is based on the efficient quadrature approximation to a certain integral representation of $\mathbb{L}^{-1}$ (see section 4).

The rest of the paper is organized as follows. Section 2 presents a sketch on multilinear algebra including the description of the canonical and Tucker models. In section 3 we consider approximation of function-related tensors. Section 4 discusses the tensor-product decomposition of multidimensional integral operators. In section 5 we consider data-sparse structured approximation of negative powers of elliptic operators. The appendix collects auxiliary results about the Sinc-approximation method.

2. Structured tensor-product decompositions.

2.1. Canonical and Tucker models. A dth order tensor $\mathcal{A} = [a_{i_1 \ldots i_d}] \in \mathbb{C}^{\mathcal{I}}$ is given, defined on the product index set $\mathcal{I} = I_1 \times \cdots \times I_d$. The canonical CP model is defined by

$$\mathcal{A}_{(r)} = \sum_{k=1}^r b_k \times_1 V_k^{(1)} \times_2 \cdots \times_d V_k^{(d)} \equiv \sum_{k=1}^r b_k \bigotimes_{\ell=1}^d V_k^{(\ell)} \approx \mathcal{A}, \quad b_k \in \mathbb{C}, \tag{2.1}$$

where the Kronecker factors $V_k^{(\ell)} \in \mathbb{C}^{I_\ell}$ are unit-norm vectors. Here and in the following we use the notation $\times_\ell$ to represent the canonical tensor

$$\mathcal{U} \equiv \{u_{\mathbf{i}}\}_{\mathbf{i}\in\mathcal{I}} = b \times_1 U^{(1)} \times_2 \cdots \times_d U^{(d)} \in \mathbb{C}^{\mathcal{I}},$$

defined by $u_{i_1\dots i_d} = b \cdot u^{(1)}_{i_1} \cdots u^{(d)}_{i_d}$ with $U^{(\ell)} \equiv \{u^{(\ell)}_{i_\ell}\}_{i_\ell \in I_\ell} \in \mathbb{C}^{I_\ell}$. We make use of the multi-index notation $\mathbf{i} := (i_1, \dots, i_d) \in \mathcal{I}$.

The minimal number r in the representation (2.1) is called the Kronecker rank of a given tensor $\mathcal{A}_{(r)}$. We denote by $\boldsymbol{\mathcal{C}}_r$ the set of componentwise normalized tensors parametrized by (2.1) and by $\boldsymbol{\mathcal{C}}_{r,\perp} \subset \boldsymbol{\mathcal{C}}_r$ the corresponding subset of orthogonally decomposable tensors (i.e., the columns of matrices $\mathbf{V}^{(\ell)} = [V^{(\ell)}_1 V^{(\ell)}_2 \dots V^{(\ell)}_r]$ ($\ell = 1, \dots, d$) are orthogonal).

The Tucker model deals with the approximation

$$\mathcal{A}_{(\mathbf{r})} = \sum_{k_1=1}^{r_1} \dots \sum_{k_d=1}^{r_d} b_{k_1\dots k_d} \times_1 V^{(1)}_{k_1} \times_2 \cdots \times_d V^{(d)}_{k_d} \equiv \sum_{\mathbf{k}=1}^{\mathbf{r}} b_{\mathbf{k}} \bigotimes_{\ell=1}^{d} V^{(\ell)}_{k_\ell} \approx \mathcal{A}, \tag{2.2}$$

where the Kronecker factors $V^{(\ell)}_{k_\ell} \in \mathbb{C}^{I_\ell}$ ($k_\ell = 1, \dots, r_\ell$, $\ell = 1, \dots, d$) are complex vectors of the respective size $n_\ell = |I_\ell|$, $\mathbf{r} = (r_1, \dots, r_d)$ (the Tucker rank) and $b_{k_1\dots k_d} \in \mathbb{C}$.

Without loss of generality, we assume that the vectors $\{V^{(\ell)}_{k_\ell}\}$ are orthonormal, i.e.,

$$\left\langle V^{(\ell)}_{k_\ell}, V^{(\ell)}_{m_\ell} \right\rangle = \delta_{k_\ell, m_\ell}, \quad k_\ell, m_\ell = 1, \dots, r_\ell;\ \ell = 1, \dots, d,$$

where δ_{k_ℓ, m_ℓ} is Kronecker's delta. In the following, we denote by $\boldsymbol{\mathcal{T}}_{\mathbf{r}}$ the set of tensors parametrized by (2.2) (i.e., $\mathbf{V}^{(\ell)} = [V^{(\ell)}_1 V^{(\ell)}_2 \dots V^{(\ell)}_{r_\ell}]$ is an orthogonal matrix for $\ell = 1, \dots, d$). We use the short notation

$$\mathcal{A}_{(\mathbf{r})} = \mathcal{B} \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} \cdots \times_d \mathbf{V}^{(d)}, \tag{2.3}$$

with tensors $\mathbf{V}^{(\ell)} \in \mathbb{R}^{I_\ell \times r_\ell}$ and $\mathcal{B} = \{b_{\mathbf{k}}\} \in \mathbb{R}^{r_1 \times \dots \times r_d}$, where the latter is called the core tensor. Notice that the representation of elements $\mathcal{A} \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$ is still not unique due to the rotational uncertainty in the core tensor $\mathcal{B}$.

The decomposition (2.1) can be viewed as a special case of the Tucker model (2.2), where $r = r_1 = \cdots = r_d$ and $b_{k_1\dots k_d} = 0$ unless $k_1 = k_2 = \cdots = k_d$, i.e., only the superdiagonal of $\mathcal{B} = \{b_{\mathbf{k}}\}$ is nonzero. If we let $r = r_\ell$, $n = n_\ell$ ($\ell = 1, \dots, d$), then both the CP and Tucker models require only *drn* numbers to represent the canonical components plus r (resp., r^d) memory units for the core tensor.

The main computational problem is the approximation of a given higher-order tensor $\mathcal{A}_0$ in a certain set of low-rank structured tensors $\boldsymbol{\mathcal{S}}$. In particular, $\boldsymbol{\mathcal{S}}$ may be one of the classes $\boldsymbol{\mathcal{T}}_{\mathbf{r}}$, $\boldsymbol{\mathcal{C}}_r$, or $\boldsymbol{\mathcal{C}}_{r,\perp}$ mentioned above. There are *algebraic*, *analytically based*, and *combined strategies* for computing a Kronecker tensor-product decomposition of a higher-order tensor.

Algebraic methods are the most general ones. The common approach is to derive the components of $\mathcal{A}_{(\mathbf{r})}$ (resp., $\mathcal{A}_{(r)}$) by straightforward minimization of the quadratic *cost functional* $f(\mathcal{A}) := \|\mathcal{A} - \mathcal{A}_0\|^2$,

$$\mathcal{A}_{(\mathbf{r})} = \operatorname{argmin} \|\mathcal{A} - \mathcal{A}_0\|^2, \tag{2.4}$$

over all rank-$\mathbf{r}$ (resp., rank-r) tensors $\mathcal{A} \in \boldsymbol{\mathcal{S}}$. Here and in the following we make use of the Frobenius (energy) norm $\|\mathcal{A}\| := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$ induced by the inner product $\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{(i_1,\dots,i_d) \in \mathcal{I}} a_{i_1\dots i_d} b_{i_1\dots i_d}$ (the maximum-norm is defined by $\|\mathcal{A}\|_\infty := \max_{\mathbf{i} \in \mathcal{I}} |a_{\mathbf{i}}|$). The difficulties in the rigorous analysis and efficient implementation

of the minimization process are due to (a) multiple local minima of the cost functional, (b) degeneracy of a minimizing sequence (in the case of the CP model), and (c) high-dimensional nonlinear optimization.

Analytically based representation methods are efficient for a special class of *function-related* operators/tensors. *Combined* methods are designed to take advantage of both algebraic and analytic approaches and, at the same time, to relax their limitations (cf. [24]).

In the case of the canonical decomposition one can find local minima of (2.4) via Newton-type algorithms applied to the Lagrange equation corresponding to the unconstrained minimization problem: Find $\mathcal{A} \in \boldsymbol{\mathcal{C}}_r$ and the Lagrange multipliers $\lambda^{(k,\ell)} \in \mathbb{R}$ such that

$$\langle \mathcal{A} - \mathcal{A}_0, \mathcal{A} - \mathcal{A}_0 \rangle + \sum_{k=1}^{r} \sum_{\ell=1}^{d} \lambda^{(k,\ell)} \left(\|V_k^{(\ell)}\|^2 - 1 \right) \to \min. \tag{2.5}$$

The complexity of one Newton step may be estimated at least by $O(dr^2 n + r^d)$.

As a second common approach, one can resort to an alternating least-squares (ALS) algorithm which is as follows: let $B = \operatorname{diag}\{b_1, \dots, b_d\}$ in (2.1) and assume that all matrices $\mathbf{V}^{(\ell)}$, $\ell \neq m$, are fixed. Then (2.4) is a quadratic expression in the components of the matrix $\mathbf{V}^{(m)} \cdot B$; hence we obtain a classical least-squares problem. To drive the solution toward local minima, an ALS iteration repeats this procedure for each component $m = 1, \dots, d$ until convergence (or termination). The components of $B = \operatorname{diag}\{b_1, \dots, b_d\}$ are obtained by normalization of the columns $V_k^{(m)}$ $(k = 1, \dots, r)$. An alternative method to compute the canonical decomposition was introduced in [5].

In general, the convergence analysis of both Newton's and ALS schemes is still an open question. In some cases, the lack of robust convergence of such nonlinear iterations is due to the already mentioned effect of degeneracy of a minimizing sequence (if $d > 2$, the corresponding set $\boldsymbol{\mathcal{C}}_r$ of structured tensors is no longer closed).

Notice that in the case of orthogonally decomposable tensors in $\boldsymbol{\mathcal{C}}_{r,\perp}$ the incremental rank-1 approximation algorithm correctly computes its CP representation (cf. [25, 30]).

We refer to [4] concerning orthogonal rank-$(r_1, \dots, r_d)$ Tucker decomposition. *Analytically based and combined* methods will be the focus of the present paper.

2.2. Canonical HKT matrix decomposition. We consider the representation problem for a class of real-valued square matrices related to discrete multidimensional operators posed in $\mathbb{R}^d$, such that $A \in \mathbb{R}^{N \times N}$, $N = n^d$. In general, such matrices can be interpreted as high-order fully populated tensors, which makes the standard matrix arithmetic almost nonfeasible. To overcome this difficulty, one needs numerically tractable data-sparse representations of high-dimensional tensors.

The HKT format as proposed in [19, 17] reads

$$A = \sum_{k=1}^{r} b_k V_k^{(1)} \otimes \cdots \otimes V_k^{(d)}, \quad b_k \in \mathbb{R}, \tag{2.6}$$

where the Kronecker factors $V_k^{(\ell)} \in \mathbb{R}^{n \times n}$ are $\mathcal{H}$-matrices (see [12, 13, 14, 15, 16] for the definition, approximation properties, and applications of $\mathcal{H}$-matrices). We recall that the Kronecker product of matrices $A \otimes B$ is defined as a block matrix $[a_{ij}B]$, provided that $A = [a_{ij}]$. The operation "$\otimes$" can be applied to arbitrary rectangular

matrices (in particular, to row or column vectors) and in the multifactor version as in (2.6).

We write $A \in \mathrm{HKT}_{(r,s)}$ if A is of the form (2.6) and if the matrices $V_k^{(\ell)}$ have a hierarchical block partitioning (independent of k) with blocks of rank at most s. The minimal number of Kronecker-product terms r involved is referred to as the *Kronecker rank*.

Approximations of function-related matrices by matrices of the form (2.6) were first studied in [19, 28]. The main result of these papers are estimates of the form $r = O(\log^2 \varepsilon)$ and $r = O(|\log \varepsilon| \log n)$, where ε is the prescribed approximation accuracy. However, if there is no structure in the Kronecker factors, then the storage is $O(drn^2)$, while the matrix-times-matrix complexity is $O(dr^2n^3)$, which may be far from being satisfactory. A possible remedy is the hierarchical ($\mathcal{H}$-matrix) approximation to the Kronecker factors (HKT approximations) with the advantage of rigorously proved existence theorems [19] with estimates of the form $r = O(\log^2 \varepsilon)$, $s = O(\log \varepsilon^{-1})$ (under certain assumptions on the origin of the matrices).

If $A \in \mathrm{HKT}_{(r,s)}$, then only the $V_k^{(\ell)}$ needs to be stored. Since, by definition, they have the $\mathcal{H}$-format, we arrive at the following complexity bounds (the linear complexity would be $O(n^d)$):

- the storage for A is $O(drsn \log n)$, indicating the *sublinear complexity*;
- multiplication of A by a rank-r_1 vector $x = \sum_{k=1}^{r_1} b_k x_k^{(1)} \otimes \cdots \otimes x_k^{(d)}$ requires $O(drr_1 sn \log n)$ operations;
- the complexity of the matrix-matrix multiplication is $O(dr^2 s^2 n \log^q n)$.

In this paper we discuss existence results for the low Kronecker rank approximation to a class of discrete integral operators.

2.3. General rank-$(r_1, \ldots, r_d)$ matrix decomposition. Let $A \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ be a real-valued matrix defined on the index set $\mathcal{I} := I_1 \times \cdots \times I_d$ with $I_\ell = \{1, \ldots, n_\ell\}$. A matrix A can be multiplied by a vector $X \in \mathbb{R}^{\mathcal{I}}$, so that $AX \in \mathbb{R}^{\mathcal{I}}$. The generalization to the case $A \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$, $X \in \mathbb{R}^{\mathcal{J}}$, is straightforward.

A matrix A (resp., a vector X) can also be regarded as a dth order tensor $\mathcal{A} \in \mathbb{R}^{I_1^2 \times \cdots \times I_d^2}$ (resp., $X \in \mathbb{R}^{I_1 \times \cdots \times I_d}$).

DEFINITION 2.1. *We introduce the following rank-$(r_1, \ldots, r_d)$ tensor-product matrix format:*

$$\mathcal{A} = \sum_{k_1=1}^{r_1} \cdots \sum_{k_d=1}^{r_d} b_{k_1 \ldots k_d} V_{k_1}^{(1)} \otimes \cdots \otimes V_{k_d}^{(d)} \in \mathbb{R}^{I_1^2 \times \cdots \times I_d^2}, \tag{2.7}$$

where the Kronecker factors $V_{k_\ell}^{(\ell)} \in \mathbb{R}^{I_\ell \times I_\ell}$, $k_\ell = 1, \ldots, r_\ell$, $\ell = 1, \ldots, d$, are matrices of a certain structure (say, $\mathcal{H}$-matrix, wavelet based format, Toeplitz/circulant, low-rank, etc.). Here $\mathbf{r} = (r_1, \ldots, r_d)$ is again called the Kronecker rank.

The matrix representation by the format (2.7) is a model reduction, which is a generalization of the low-rank approximation of matrices, corresponding to the case $d = 2$.

Remark 2.2. The matrix representation (2.7) is reminiscent of the Tucker decomposition of multidimensional tensors (cf. (2.2)), while (2.6) comply with the CP model (cf. (2.1)). With the help of the so-called n-mode tensor-matrix product (cf. [4]), we introduce the short notation

$$\mathcal{A} = \mathcal{B} \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} \cdots \times_d \mathbf{V}^{(d)} \equiv \mathcal{B} \times_{\mathbf{r}} \{\mathbf{V}\}, \tag{2.8}$$

with tensors $\mathbf{V}^{(\ell)} \in \mathbb{R}^{I_\ell \times I_\ell \times r_\ell}$ and $\mathcal{B} = \{b_{\mathbf{k}}\} \in \mathbb{R}^{r_1 \times \cdots \times r_d}$, where the latter is the core tensor. We denote matrices by uppercase letters, e.g., A, and tensors by calligraphic letters, e.g., $\mathcal{B}$. In addition, we set $\mathbf{A} = [A_1 A_2 \dots]$, where A_i is the ith column matrix/vector of $\mathbf{A}$, e.g., $\mathbf{V}^{(\ell)} = [V_1^{(\ell)} V_2^{(\ell)} \dots V_{r_\ell}^{(\ell)}]$.

Similarly to the class of tensors $\boldsymbol{\mathcal{T}}_{\mathbf{r}}$ in the Tucker model, i.e., if the components $V_{k_\ell}^{(\ell)} \in \mathbb{R}^{I_\ell}$ in (2.2) are mutually orthogonal vectors of an arbitrary structure, we introduce the notation $A \in \boldsymbol{\mathcal{T}}_{\mathcal{H},\mathbf{r}}$ for the tensor-product matrix format (2.8) with the canonical components having hierarchical structure.

Clearly, we have

$$\boldsymbol{\mathcal{C}}_r = \boldsymbol{\mathcal{T}}_{\mathbf{r}} \quad \text{if } r = 1; \quad \boldsymbol{\mathcal{C}}_r^{\perp} \subset \boldsymbol{\mathcal{T}}_{\mathbf{r}} \quad \text{if } \mathbf{r} = (r, \dots, r).$$

In general, the CP decomposition (2.1) cannot be retrieved by rotation and "diagonal truncation" of the Tucker model. However, it is possible for orthogonally decomposable tensors in $\boldsymbol{\mathcal{C}}_r^{\perp}$.

We simplify the complexity analysis and set $r_\ell = r$, $n_\ell = n$ $(\ell = 1, \dots, d)$; the general case can be treated completely similarly. A dth order tensor is called supersymmetric if it is invariant under arbitrary permutations of indices in $\{1, \dots, d\}$.

LEMMA 2.3 (see [24]). *The storage cost for $A \in \boldsymbol{\mathcal{T}}_{\mathcal{H},\mathbf{r}}$ is estimated by $O(drsn \log n) + r^d$, while for the supersymmetric tensor we arrive at the memory consumption $O(rsn \log n) + \frac{r^d}{d}$.*

Multiplication by a rank-r_0 vector requires $O(drr_0sn \log n)$ operations. Let $A_1, A_2 \in \boldsymbol{\mathcal{T}}_{\mathcal{H},\mathbf{r}}$. Then both $A_1 A_2$ and the Hadamard matrix product $A_1 \odot A_2$ can be computed and stored in $O(dr^2 s^2 n \log n) + r^{2d}$ operations.

Proof. The storage requirement for A is trivial. Let $\mathcal{X} = \times_1 x_1 \times_2 \cdots \times_d x_d$ with $x_\ell \in \mathbb{R}^{I_\ell}$. Then

$$\mathcal{A}\mathcal{X} \equiv \mathcal{B} \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} \cdots \times_d \mathbf{V}^{(d)} \mathcal{X} = \mathcal{B} \times_1 \mathbf{V}^{(1)} x_1 \times_2 \mathbf{V}^{(2)} x_2 \cdots \times_d \mathbf{V}^{(d)} x_d$$

implies the second assertion. Now we set

$$\mathcal{A}_1 = \mathcal{B} \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} \cdots \times_d \mathbf{V}^{(d)}, \quad \mathcal{A}_2 = \mathcal{C} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_d \mathbf{U}^{(d)}$$

to obtain the representation

$$\mathcal{A}_1 \mathcal{A}_2 = \left(\sum_{\mathbf{k}=1}^{\mathbf{r}} b_{\mathbf{k}} \bigotimes_{\ell=1}^{d} V_{k_\ell}^{(\ell)} \right) \left(\sum_{\mathbf{m}=1}^{\mathbf{r}} c_{\mathbf{m}} \bigotimes_{\ell=1}^{d} U_{m_\ell}^{(\ell)} \right) = \sum_{\mathbf{k}=1}^{\mathbf{r}} \sum_{\mathbf{m}=1}^{\mathbf{r}} b_{\mathbf{k}} c_{\mathbf{m}} \bigotimes_{\ell=1}^{d} V_{k_\ell}^{(\ell)} U_{m_\ell}^{(\ell)},$$

which includes dr^2 canonical components, and where the core tensor $\{b_{\mathbf{k}} c_{\mathbf{m}}\}$ has $(r^2)^d = r^{2d}$ entries. Analogously, for the Hadamard product we have

$$\mathcal{A}_1 \odot \mathcal{A}_2 = \left(\sum_{\mathbf{k}=1}^{\mathbf{r}} b_{\mathbf{k}} \bigotimes_{\ell=1}^{d} V_{k_\ell}^{(\ell)} \right) \odot \left(\sum_{\mathbf{m}=1}^{\mathbf{r}} c_{\mathbf{m}} \bigotimes_{\ell=1}^{d} U_{m_\ell}^{(\ell)} \right) = \sum_{\mathbf{k}=1}^{\mathbf{r}} \sum_{\mathbf{m}=1}^{\mathbf{r}} b_{\mathbf{k}} c_{\mathbf{m}} \bigotimes_{\ell=1}^{d} V_{k_\ell}^{(\ell)} \odot U_{m_\ell}^{(\ell)}$$

and take into account that the Hadamard product of two $\mathcal{H}$-matrices has linear-logarithmic cost (cf. [22]). This completes our proof. ∎

In the case of the CP decomposition (2.6) we introduce the notation $A \in \boldsymbol{\mathcal{C}}_{\mathcal{I}\times\mathcal{I},r}$ or more specifically $A \in \boldsymbol{\mathcal{C}}_{\mathcal{H},r}$ if the canonical components are matrices of the general or $\mathcal{H}$-matrix structure, respectively (in particular, we can now identify $HKT_{(r,s)} \equiv \boldsymbol{\mathcal{C}}_{\mathcal{H},r}$).

3. Approximation of function generated tensors. In this section we discuss the low Kronecker rank approximation of a special class of higher-order tensors related to certain "discretizations" of multivariate functions, which will be called function-generated tensors (FGTs). They directly arise from

(a) Nyström/collocation/Galerkin discretizations of integral operators;

(b) the approximation to some analytic matrix-valued functions.

3.1. Basic definitions. In the following we define FGTs corresponding to the Nyström/collocation and Galerkin discretizations.

In the case of an interpolation method, we let $\mathcal{I}_\ell = I_{\ell,1} \times \cdots \times I_{\ell,p}$ be the product index set, where we use multi-indices $\mathbf{i}_\ell = (i_{\ell,1}, \ldots, i_{\ell,p}) \in \mathcal{I}_\ell$ $(\ell = 1, \ldots, d)$ with the components $i_{\ell,m} \in \{1, \ldots, n\}$ $(m = 1, \ldots, p)$. Furthermore, let ω_ℓ be a uniform rectangular grid on $\Pi := [a_0, b_0]^p$, $a_0, b_0 > 0$, indexed by $\mathcal{I}_\ell$, and let $\{\zeta^{(1)}_{\mathbf{i}_1}, \ldots, \zeta^{(d)}_{\mathbf{i}_d}\}$ with $\mathbf{i}_\ell \in \mathcal{I}_\ell$ $(\ell = 1, \ldots, d)$ be a set of collocation points living on the tensor-product lattice $\omega_{\mathbf{d}} := \omega_1 \times \cdots \times \omega_d$ in a hypercube $\Omega := \Pi^d \subset \mathbb{R}^{\mathbf{d}}$ with $\mathbf{d} = dp$ so that $(\mathbf{i}_1, \ldots, \mathbf{i}_d) \in \mathcal{I}^{(\mathbf{d})} := \mathcal{I}_1 \times \cdots \times \mathcal{I}_d$. We also define $|\mathbf{i}_\ell| = \max_{m \le p} |i_{\ell,m}|$ and similarly for $|\mathbf{i}|$, $\mathbf{i} = (i_1, \ldots, i_d)$. In our applications we have $d \ge 2$ with some fixed $p = 1, 2, 3$.

DEFINITION 3.1 (collocation case, FGT(C)). *Given a multivariate function $g : \Omega \to \mathbb{R}$, we introduce the collocation-type FGT of order d by*

$$\mathcal{A} \equiv \mathcal{A}(g) := [a_{\mathbf{i}_1 \ldots \mathbf{i}_d}] \in \mathbb{R}^{\mathcal{I}_1 \times \cdots \times \mathcal{I}_d} \quad \textit{with} \quad a_{\mathbf{i}_1 \ldots \mathbf{i}_d} := g(\zeta^{(1)}_{\mathbf{i}_1}, \ldots, \zeta^{(d)}_{\mathbf{i}_d}). \tag{3.1}$$

In the case of Galerkin schemes we make use of tensor-product test functions

$$\phi^{\mathbf{i}}(x_1, \ldots, x_d) = \prod_{\ell=1}^{d} \phi^{i_\ell}_\ell(x_\ell), \quad \mathbf{i} = (i_1, \ldots, i_d) \in \mathbb{R}^{\mathcal{I}_1 \times \cdots \times \mathcal{I}_d},\ i_\ell \in I_n := \{1, \ldots, n\}, \tag{3.2}$$

and $\psi^{\mathbf{j}}$ with $\mathbf{j} = (j_1, \ldots, j_d) \in \mathbb{R}^{\mathcal{J}_1 \times \cdots \times \mathcal{J}_d}$, $j_\ell \in I_n$, of similar product form.

DEFINITION 3.2 (Galerkin case, FGT(G)). *Given a multivariate function $g : \Omega \times \Omega \to \mathbb{R}$ with $\Omega \subset \mathbb{R}^d$, and a tensor-product basis set* (3.2)*, we let $p = 2$, $\zeta^{(\ell)} = (x_\ell, y_\ell)$, $\mathbf{m}_\ell = (i_\ell, j_\ell) \in \mathcal{M}_\ell := I_{\ell,1} \times I_{\ell,2}$ and introduce the Galerkin-type dth order FGT by $\mathcal{A} \equiv \mathcal{A}(g) := [a_{\mathbf{m}_1 \ldots \mathbf{m}_d}] \in \mathbb{R}^{\mathcal{M}_1 \times \ldots \times \mathcal{M}_d}$ with*

$$a_{\mathbf{m}_1 \ldots \mathbf{m}_d} := \int_{\Omega \times \Omega} g(\zeta^{(1)}, \ldots, \zeta^{(d)}) \phi^{\mathbf{i}}(x_1, \ldots, x_d) \psi^{\mathbf{j}}(y_1, \ldots, y_d) dx dy. \tag{3.3}$$

In the numerical calculations involving integral operators (e.g., arising from the Hartree–Fock or Boltzmann equations), n may vary from several hundreds to several thousands, and therefore, for $d \ge 3$, a naive "entrywise" representation to the tensor $\mathcal{A}$ in (3.1) amounts to substantial computer resources (at least of the order $O(n^{dp})$).

3.2. Kronecker rank in CP decomposition. We recall that CP-type decompositions like (2.1) (or (2.6) in the matrix case) can be derived by using a corresponding separable expansion of the generating function g (see [17, 19] for more details). Assume that we are given a set of functions $\{\Phi^{(\ell)}_k : \mathbb{R}^p \to \mathbb{R}\}$ $(\ell = 1, \ldots, d)$ with the following property.

PROPOSITION 3.3. *Suppose that a multivariate function $g : \Omega \to \mathbb{R}$ can be approximated by a separable expansion*

$$g_r(\boldsymbol{\zeta}) := \sum_{k=1}^{r} \mu_k \Phi^{(1)}_k(\zeta^{(1)}) \cdots \Phi^{(d)}_k(\zeta^{(d)}) \approx g(\boldsymbol{\zeta}), \quad \zeta^{(\ell)} \in \mathbb{R}^p, \quad \ell = 1, \ldots, d, \tag{3.4}$$

where $\mu_k \in \mathbb{R}$ *and* $\Phi_k^{(\ell)} : \Pi \to \mathbb{R}$. *Then the FGT(C) defined by the CP decomposition* (2.1) *via* $\mathcal{A}_{(r)} := \mathcal{A}(g_r)$ *as in Definition* 3.1 *with*

$$V_k^{(\ell)} = \{\Phi_k^{(\ell)}(\zeta_{\mathbf{i}_\ell}^{(\ell)})\}_{\mathbf{i}_\ell \in \mathcal{I}_\ell} \in \mathbb{R}^{\mathcal{I}_\ell}, \quad b_k = \mu_k, \tag{3.5}$$

and the FGT(G), corresponding to the choice

$$V_k^{(\ell)} = \int \Phi_k^{(\ell)}(\zeta_{\mathbf{i}_\ell}^{(\ell)})\phi_\ell^{i_\ell}(x_\ell)\psi_\ell^{j_\ell}(y_\ell)dx_\ell dy_\ell \in \mathbb{R}^{\mathcal{I}_\ell \times \mathcal{J}_\ell}, \quad \ell = 1, \dots, d, \quad k = 1, \dots, r, \tag{3.6}$$

both provide the error estimate $\|\mathcal{A}(g) - \mathcal{A}_{(r)}(g_r)\|_\infty \le C\|g - g_r\|_{L^\infty(\Omega)}$, *where* $C = 1$ *in the FGT(C) case.*

Proof. The analysis for FGT(C) is presented in [17]. In the Galerkin case we readily obtain

$$\begin{aligned} |a_{\mathbf{m}_1 \dots \mathbf{m}_d} - a^{(r)}_{\mathbf{m}_1 \dots \mathbf{m}_d}| &= \left| \int_{\Omega \times \Omega} (g(x,y) - g_r(x,y))\phi^{\mathbf{i}}(x)\psi^{\mathbf{j}}(y)dxdy \right| \\ &\le \|g - g_r\|_{L^\infty(\Omega)} \int_{\Omega \times \Omega} \left|\phi^{\mathbf{i}}(x)\psi^{\mathbf{j}}(y)\right| dxdy; \end{aligned}$$

then the result follows. □

In computationally efficient algorithms the separation rank r is supposed to be as small as possible, while the set of functions $\{\Phi_k^{(\ell)} : \mathbb{R}^p \to \mathbb{R}\}$ can be fixed a priori or chosen adaptively to the problem.

Though in general the construction of a decomposition (3.4) with small separation rank r is a complicated numerical task, in many interesting applications efficient approximation methods are available. In particular, for a class of multivariate functions (say, certain shift-invariant *Green's kernels* in $\mathbb{R}^d$) it is possible to obtain a dimensionally independent Kronecker rank estimate $r = O(\log n |\log \varepsilon|)$ based on *sinc*-quadrature methods or an approximation by exponential sums (cf. case study examples in [2, 17, 23]).

In the rest of this section we discuss the constructive CP decomposition of FGTs applied to a *general class of generating functions* characterized in terms of their Laplace transform. The construction is based on *sinc-approximation methods*.

We consider a class of multivariate functions $g : \mathbb{R}^d \to \mathbb{R}$ parametrized by $g(\boldsymbol{\zeta}) = G(\rho)$ with $\rho = \rho_1(\zeta_1) + \dots + \rho_d(\zeta_d) > 0$, $\rho_\ell : \mathbb{R}^p \to \mathbb{R}_+$ (with small $p \in \mathbb{N}_{\ge 1}$), where a univariate function $G : \mathbb{R}_+ \to \mathbb{R}$ can be represented via the Laplace transform

$$G(\rho) = \int_{\mathbb{R}_+} \mathcal{G}(\tau)e^{-\rho\tau}d\tau.$$

Now the FGT(G) approximation corresponds to $p = 2$, $\zeta_\ell = (x_\ell, y_\ell)$. Without loss of generality, we suppose that $\phi_\ell^{i_\ell}(\cdot) = \phi(\cdot + (i_\ell - 1)h)$ $(\ell = 1, \dots, d)$ with a single scaling function ϕ, where $h > 0$ is the mesh parameter, and the same for $\psi_\ell^{j_\ell}(\cdot)$. We also simplify and set $\rho_\ell = \rho_0(x_\ell, y_\ell)$ $(\ell = 1, \dots, d)$ and, moreover, $\rho_0 : [a,b]^2 \to [a_1, b_1] \subset \mathbb{R}_{>0}$, while $\rho \in [a_d, b_d] \subset \mathbb{R}_{>0}$. The more general multilevel setting (say, corresponding to a wavelet basis) can be analyzed completely similarly. For each $i, j \in I_n$, we introduce the parameter-dependent function

$$\Psi_{i,j}(\tau) := \int_{\mathbb{R}^2} e^{-\tau\rho_0(x,y)}\phi(x + (i-1)h)\psi(y + (j-1)h)dxdy, \quad \tau \ge 0,$$

as well as an auxiliary function $f_I(\tau) := \mathcal{G}(\tau)e^{-\rho\tau}$.

THEOREM 3.4 (FGT(C) approximation). *Assume that*

(a) $\mathcal{G}(\tau)$ *has an analytic extension* $\mathcal{G}(w)$, $w \in \Omega_{\mathcal{G}}$, *into a certain domain* $\Omega_{\mathcal{G}} \subset \mathbb{C}$ *which can be mapped conformally onto the strip* D_δ *(see the appendix), such that* $w = \phi(z)$, $z \in D_\delta$, *and* $\phi^{-1} : \Omega_{\mathcal{G}} \to D_\delta$;

(b) *for each* $\rho \in [a_d, b_d]$ *with* $a_d > 0$, *the function* $f(z) := \phi'(z) f_I(\phi(z))$ *belongs to the Hardy space* $H^1(D_\delta)$ *with* $N(f, D_\delta) < \infty$ *uniformly in* ρ;

(c) $f(t)$, $t \in \mathbb{R}$, *has* (c1) *exponential or* (c2) *hyperexponential decay as* $t \to \pm\infty$.

Then, for each $M \in \mathbb{N}_+$, *the FGT(C),* $\mathcal{A}(g)$, *defined on* $[a, b]^d$ *allows an exponentially convergent supersymmetric CP decomposition* $\mathcal{A}_{(r)} \in \boldsymbol{\mathcal{C}}_r$ *with* $V_k^{(\ell)}$ *as in* (3.5), *where* $\Phi_k^{(\ell)}(\zeta^{(\ell)}) = e^{-a_k \zeta^{(\ell)}}$ $(\ell = 1, \ldots, d)$, *and where* μ_k, a_k *are given by applying the sinc-quadrature* (6.2) *with* $f(t) = \phi'(t) f_I(\phi(t))$, *such that we have*

$$\|\mathcal{A}(g) - \mathcal{A}_{(r)}\|_\infty \le Ce^{-\alpha M^\nu} \quad \text{with } r = 2M + 1, \tag{3.7}$$

where $\nu = \frac{1}{2}$, $\alpha = \sqrt{2\pi\delta b}$ *in the case* (c1) *and with* $\nu = 1$, $\alpha = \frac{2\pi\delta b}{\log(2\pi a M/b)}$ *in the case* (c2).

(FGT(G) approximation) *Assume that* (a) *holds and for each* $\rho \in [a_d, b_d]$ *and* $(\mathbf{i}, \mathbf{j}) \in \mathcal{I} \times \mathcal{J}$:

(b′) *The transformed integrand* $f(z) := \phi'(z)\mathcal{G}(\phi(z)) \prod_{\ell=1}^d \Psi_{i_\ell j_\ell}(\phi(z))$ *belongs to the Hardy space* $H^1(D_\delta)$ *with* $N(f, D_\delta) < \infty$ *uniformly in* ρ; *item* (c) *holds.*

Then, for each $M \in \mathbb{N}$, *the FGT(G),* $\mathcal{A}(g)$, *defined on* $[a, b]^d$ *allows a supersymmetric CP decomposition* $\mathcal{A}_{(r)} \in \boldsymbol{\mathcal{C}}_r$ *with* $V_k^{(\ell)}$ *as in* (3.6) *that yields the error estimate* (3.7).

Proof. In the FGT(C) case, we directly apply the sinc-quadrature theory to the transformed integrand $f(z)$ to obtain $T_M(f, \mathfrak{h}) := \mathfrak{h} \sum_{k=-M}^M f(k\mathfrak{h}) \approx G(\rho)$ (cf. the appendix) with

$$|G(\rho) - T_M(\rho)| \le Ce^{-\alpha M^\nu}, \quad \rho \in [a_d, b_d],$$

and with the respective α, ν. Combining this estimate with Proposition 3.3 and taking into account the separability property of the exponential proves the first assertion.

To prove the FGT(G) case, we notice that by definition

$$a_{\mathbf{ij}} = \int_{\mathbb{R}_+} \mathcal{G}(\tau) \prod_{\ell=1}^d \Psi_{i_\ell j_\ell}(\tau) d\tau \quad \text{for } (\mathbf{i}, \mathbf{j}) \in \mathcal{I} \times \mathcal{J}.$$

We again apply the sinc-quadrature to the transformed integrand $f(z)$ and obtain the exponential convergence as in the case of FGT(C) approximation. Notice that our quadrature does not depend on the index $(\mathbf{i}, \mathbf{j})$, which completes the proof. □

Theorem 3.4 proves the existence of a CP decomposition to the FGT $\mathcal{A}(g)$ with the Kronecker rank $r = O(|\log \varepsilon| \log n)$ (in the case (c2)) or $r = O(\log^2 \varepsilon)$ (in the case (c1)).

3.3. Rank estimates for rank-$(r_1, \ldots, r_d)$ Tucker model. For the class of applications based on separation via tensor-product interpolation, the CP model typically leads to the Kronecker rank estimate $r_{CP} = r^d$ with $r = O(\log n |\log \varepsilon|)$, where the dimensional parameter d gets into the exponent. In such cases one can apply the rank-$(r_1, \ldots, r_d)$ Tucker decomposition instead of the rank-r_{CP} canonical model.

The next lemma shows that the error of the Tucker decomposition is directly related to the error of the separable approximation of the generating function.

LEMMA 3.5 (see [24]). *Let $g : \Omega \to \mathbb{R}$ be approximated by a separable expansion*
(3.8)
$$g_{\mathbf{r}}(\boldsymbol{\zeta}) := \sum_{k_1=1}^{r_1} \cdots \sum_{k_d=1}^{r_d} b_{k_1\dots k_d} \Phi^{(1)}_{k_1}(\zeta^{(1)}) \cdots \Phi^{(d)}_{k_d}(\zeta^{(d)}) \approx g, \quad \zeta^{(\ell)} \in \mathbb{R}^p, \quad 1 \le \ell \le d,$$

where $b_{k_1\dots k_d} \in \mathbb{R}$. Then both the FGT(C) of the form $\mathcal{A}_{(\mathbf{r})} := \mathcal{A}(g_{\mathbf{r}}) \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$ generated by $g_{\mathbf{r}}$ with

$$V^{(\ell)}_{k_\ell} = \{\Phi^{(\ell)}_{k_\ell}(\zeta^{(\ell)}_{\mathbf{i}_\ell})\}_{\mathbf{i}_\ell \in \mathcal{I}_\ell} \in \mathbb{R}^{\mathcal{I}_\ell} \tag{3.9}$$

and the FGT(G), corresponding to the choice
(3.10)
$$V^{(\ell)}_{k_\ell} = \int \Phi^{(\ell)}_{k_\ell}(\zeta^{(\ell)}_{\mathbf{i}_\ell}) \phi^{i_\ell}_\ell(x_\ell) \psi^{j_\ell}_\ell(y_\ell) dx_\ell dy_\ell \in \mathbb{R}^{\mathcal{I}_\ell \times \mathcal{J}_\ell}, \quad \ell = 1, \dots, d, \quad k_\ell = 1, \dots, r_\ell,$$

provide the error estimate $\|\mathcal{A}(g) - \mathcal{A}_{(\mathbf{r})}(g_{\mathbf{r}})\|_\infty \le C\|g - g_{\mathbf{r}}\|_{L^\infty(\Omega)}$, where $C = 1$ in the FGT(C) case.

Proof. In the FGT(C) case, by the construction of $\mathcal{A}_{(\mathbf{r})}$ we have

$$\|\mathcal{A} - \mathcal{A}_{(\mathbf{r})}\|_\infty = \max_{(\mathbf{i}_1,\dots,\mathbf{i}_d) \in \mathcal{I}^{\mathbf{d}}} \left\{ \left| g(\zeta^{(1)}_{\mathbf{i}_1}, \dots, \zeta^{(d)}_{\mathbf{i}_d}) - \sum_{k_1=1}^{r_1} \cdots \right. \right.$$
$$\left. \left. \sum_{k_d=1}^{r_d} b_{k_1\dots k_d} \Phi^{(1)}_{k_1}(\zeta^{(1)}_{\mathbf{i}_1}) \cdots \Phi^{(d)}_{k_d}(\zeta^{(d)}_{\mathbf{i}_d}) \right| \right\},$$

which proves the first assertion. The Galerkin-type approximation can be analyzed as in Proposition 3.3. ∎

For a class of analytic functions with point singularities the expansion (3.8) can be derived via tensor-product *Sinc*-interpolation, which is motivated by various favorable features of the Sinc-basis in $L^2(\mathbb{R})$ (cf. [22]).

COROLLARY 3.6. *Assume that $g(\zeta)$ satisfies the requirements for the tensor-product sinc-interpolation (cf. the appendix). Then the FGT(C), $\mathcal{A}(g)$, allows an exponentially convergent rank-$(r,\dots,r)$ decomposition $\mathcal{A}_{(\mathbf{r})} \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$ with $V^{(\ell)}_{k_\ell}$ as in* (3.9)*, where $\Phi^{(\ell)}_{k_\ell}(\zeta^{(\ell)}) = \operatorname{sinc}(-a_{k_\ell}\zeta^{(\ell)})$ $(\ell = 1,\dots,d)$, and where $b_{\mathbf{k}}$ are represented via the sinc-interpolation* (6.3) *applied to the function g, such that*

$$\|\mathcal{A}(g) - \mathcal{A}_{(\mathbf{r})}\|_\infty \le C(1 + \log M)^d e^{-\alpha M^\nu} \quad \text{with } r = 2M + 1, \tag{3.11}$$

with $\nu = \frac{1}{2}$, $\alpha = \sqrt{2\pi\delta b}$ in the case (c1) *and with $\nu = 1$, $\alpha = \frac{2\pi\delta b}{\log(2\pi a M/b)}$ in the case* (c2) *as in Theorem* 3.4*. A similar result holds in the FGT(G) case.*

Proof. We apply Lemma 3.5, yielding an exponential error bound for the tensor-product sinc-interpolation (cf. the appendix), which proves the first assertion. The FGT(G) case can be analyzed similarly. ∎

The error estimate (3.11) yields $\max_\ell r_\ell = O(|\log\varepsilon|\delta^{-1})$. In some cases we get the estimate $\delta^{-1} = O(\log n)$ (cf. [17]).

4. Structured approximation of integral operators.

4.1. Canonical-HKT decomposition in $\mathbb{R}^d$. The principal ingredient in the HKT representation of integral operators in many spatial dimensions is a separable approximation of the multivariate function representing the kernel of the operator. Given the integral operator $\mathbb{G} : L^2(\Omega) \to L^2(\Omega)$ in $\Omega := [0,1]^d \subset \mathbb{R}^d$, $d \geq 2$,

$$(\mathbb{G}u)(x) := \int_\Omega g(x,y)u(y)\mathrm{d}y, \quad x, y \in \Omega,$$

with some shift-invariant kernel function $g(x,y) = g(|x-y|)$, which, therefore, can be represented in the form

$$g(x,y) = G(\zeta_1, \ldots, \zeta_d) \equiv g\left(\sqrt{\zeta_1^2 + \cdots + \zeta_d^2}\right),$$

where $\zeta_\ell = |x_\ell - y_\ell| \in [0,1]$, $\ell = 1, \ldots, d$.

In the case of a continuous kernel function one can apply the so-called Nyström approximation with respect to the tensor-product grid $\omega_h \times \omega_h \in \Pi^d \times \Pi^d$,

$$(\mathbb{G}u) \approx \sum_{y_{\mathbf{j}} \in \omega_h} g(x_{\mathbf{i}}, y_{\mathbf{j}})u(y_{\mathbf{j}}), \quad x_{\mathbf{i}} \in \omega_h.$$

Hence the previous analysis for the FGT(C) approximation directly applies to the arising stiffness matrix $A = \{g(x_{\mathbf{i}}, y_{\mathbf{j}})\}$.

In the presence of diagonal singularities, a separable approximation may be easier for the following modified kernels. With some fixed $0 \leq \alpha_0 < 1$, we introduce the auxiliary function

$$F(\zeta_1, \ldots, \zeta_d) := (\zeta_1 \cdots \zeta_{d-1})^{\alpha_0} G(\zeta_1, \ldots, \zeta_d). \tag{4.1}$$

In this section we suppose that the multivariate function $F : \mathbb{R}^d \to \mathbb{R}$ can be approximated by a separable expansion

$$F_r(\zeta_1, \ldots, \zeta_d) := \sum_{k=1}^{r} \Phi_k^{(1)}(\zeta_1) \cdots \Phi_k^{(d)}(\zeta_d) \approx F, \tag{4.2}$$

where the set of functions $\{\Phi_k^{(\ell)} : \ell = 1, \ldots, d,\ k = 1, \ldots, r\}$ with $\Phi_k^{(\ell)} : [0,1] \to \mathbb{R}$ may be fixed or can be chosen adaptively. Various methods for constructing approximations which are exponentially convergent in r are discussed in [17].

We consider a Galerkin scheme with tensor-product test functions

$$\phi^{\mathbf{i}}(x_1, \ldots, x_d) = \phi_1^{i_1}(x_1) \cdot \ldots \cdot \phi_d^{i_d}(x_d),$$
$$\mathbf{i} = (i_1, \ldots, i_d),\ i_\ell \in I_n := \{1, \ldots, n\},\ \ell = 1, \ldots, d.$$

Now we approximate the Galerkin stiffness matrix

$$A = \{(\mathbb{G}\phi^{\mathbf{i}}, \phi^{\mathbf{j}})_{L^2}\}_{\mathbf{i},\mathbf{j} \in I_n^d} \in \mathbb{R}^{N \times N}, \quad N = n^d,$$

by a matrix $A_{(r)}$ of the form (2.6), where the $V_k^{(\ell)}$ are $n \times n$ matrices given by

$$V_k^{(\ell)} = \left\{ \int_0^1 |x_\ell - y_\ell|^{-\alpha_\ell} \Phi_k^\ell(|x_\ell - y_\ell|) \phi_\ell^{i_\ell}(x_\ell) \phi_\ell^{j_\ell}(y_\ell) \mathrm{d}x_\ell \mathrm{d}y_\ell \right\}_{i_\ell, j_\ell = 1}^{n}, \quad \ell = 1, \ldots, d, \tag{4.3}$$

with $\alpha_\ell = \alpha_0$, $\ell = 1, \dots, d-1$, and $\alpha_d = 0$ (see (4.1)).

In the following we apply the approximation properties of asymptotically smooth functions. We recall that a function $g(x,y)$, $x, y \in \mathbb{R}^d$, is called asymptotically smooth if there exists $\gamma \geq 1$, and $p \in \mathbb{R}$ such that for all $x, y \in \mathbb{R}^d$, $x \neq y$, and all multi-indices α, β such that $|\alpha| + |\beta| > 0$ with $|\alpha| = \alpha_1 + \cdots + \alpha_d$, there holds

$$|\partial_x^\alpha \partial_y^\beta g(x,y)| \leq C\alpha!\beta!\gamma^{|\alpha|+|\beta|}|x-y|^{-p-|\alpha|-|\beta|}.$$

The next lemma shows that the error $\|A - A_{(r)}\|_\infty$ is directly related to the error $\|F - F_r\|_\infty$ of the separable approximation (4.2) of F (see the discussion in [19]). It also specifies sufficient conditions for $\mathcal{H}$-matrix approximability to the Kronecker factors $V_k^{(\ell)}$.

LEMMA 4.1 (see [18]). *Let* (4.2) *be valid; then for any* $\mathbf{i}, \mathbf{j} \in I_n^d$, *we have the estimate*

$$|a_{\mathbf{i},\mathbf{j}} - a^r_{\mathbf{i},\mathbf{j}}| \leq \|F - F_r\|_\infty \prod_{\ell=1}^d \left\| |x_\ell - y_\ell|^{-\alpha_\ell} \phi_\ell^{i_\ell}(x_\ell)\phi_\ell^{j_\ell}(y_\ell) \right\|_{L^1([0,1]\times[0,1])}$$

for the components of $A - A_{(r)}$. *We assume that the function*

$$g_{\ell,k}(u,v) := |u-v|^{-\alpha_\ell}\Phi_k^{(\ell)}(|u-v|), \quad (u,v) \in [0,1]^2,$$

is asymptotically smooth for $\ell = 1, \dots, d$, $k = 1, \dots, r$. *Then, for low-order piecewise polynomial basis functions,* $V_k^{(\ell)}$ *can be approximated by a rank-m* $\mathcal{H}$*-matrix* $\widetilde{V}_k^{(\ell)}$ *with an error*

$$\|V_k^{(\ell)} - \widetilde{V}_k^{(\ell)}\| \leq C\eta^m \quad \text{for some} \quad \eta < 1.$$

Proof. By construction we obtain

$$\begin{aligned}
|a_{\mathbf{i},\mathbf{j}} - a^r_{\mathbf{i},\mathbf{j}}| &= \left| \int_{\Omega\times\Omega} (F - F_r) \left(\prod_{\ell=1}^d |x_\ell - y_\ell|^{-\alpha_\ell} \right) \phi^{\mathbf{i}}(x)\phi^{\mathbf{j}}(y)\mathrm{d}x\mathrm{d}y \right| \\
&\leq \|F - F_r\|_\infty \left\| \left(\prod_{\ell=1}^d |x_\ell - y_\ell|^{-\alpha_\ell} \right) \phi^{\mathbf{i}}(x)\phi^{\mathbf{j}}(y) \right\|_{L^1(\Omega\times\Omega)} \\
&= \|F - F_r\|_\infty \prod_{\ell=1}^d \left\| |x_\ell - y_\ell|^{-\alpha_\ell} \phi_\ell^{i_\ell}(x_\ell)\phi_\ell^{j_\ell}(y_\ell) \right\|_{L^1([0,1]\times[0,1])},
\end{aligned}$$

where the last equation follows by inserting the tensor-product basis and by separating the $2d$-dimensional integral.

To prove the second statement, we note that $V_k^{(\ell)}$ given by (4.3) appears to be the exact Galerkin stiffness matrix for an integral operator with the kernel function $g_{\ell,k}(u,v)$ defined on $[0,1]\times[0,1]$. Since $g_{\ell,k}(u,v)$ is supposed to be asymptotically smooth, the result follows by the conventional theory of the $\mathcal{H}$-matrix approximation (cf. [12, 14, 15, 16]). ∎

Note that due to Lemma 4.1, $\|A - A_{(r)}\|$ can be easily estimated in, say, the Frobenius, l_2, or l_∞ matrix norms. In particular, we have

$$\|A - A_{(r)}\|_\infty \leq n^d \|F - F_r\|_\infty \prod_{\ell=1}^d \left\| |x_\ell - y_\ell|^{-\alpha_\ell} \phi_\ell^{i_\ell}(x_\ell)\phi_\ell^{j_\ell}(y_\ell) \right\|_{L^1([0,1]\times[0,1])}.$$

Several methods of separable approximations to multivariate functions are presented in [17]. In the general case, the approximability property (4.2) can be validated by using the tensor-product Sinc-interpolation. In this case the function $\Phi_k^{(\ell)}(|u-v|)$ can be proved to be asymptotically smooth. For the class of kernel functions approximated by the quadrature method or by exponential sums, the factor $\Phi_k^{(\ell)}(|u-v|)$ even appears to be globally smooth (indeed, it is the entire function).

LEMMA 4.2 (see [18]). *For both, the tensor-product Sinc-interpolation and quadrature approximation methods, the function $g_{\ell,k}(u,v)$ from Lemma 4.1 is asymptotically smooth.*

Proof. In the first case we have

$$g_{\ell,k}(u,v) = |u-v|^{-\alpha_\ell} S(k,\mathfrak{h})(\phi^{-1}(|u-v|)), \qquad u,v \in [0,1],$$

where $S(k,\mathfrak{h})$ refers for the kth Sinc-function with step-size $\mathfrak{h}$, and (cf. [17, section 2])

$$\phi^{-1}(x) = \operatorname{arsinh}\left(\operatorname{arcosh}\left(\frac{1}{x}\right)\right).$$

Since the Sinc-function $S(k,\mathfrak{h})(x)$, $x \in \mathbb{R}$, is holomorphic in x, and since the factor $|u-v|^{-\alpha_\ell}$ is asymptotically smooth, we conclude that $g_{\ell,k}(u,v)$ is also asymptotically smooth. In the case of a quadrature method, we obtain the entire function

$$\Phi_k^{(\ell)}(|u-v|) = \exp(-t_k|u-v|^2), \quad t_k > 0.$$

Then the previous argument completes the proof. □

Applying Lemmas 4.1 and 4.2 proves the existence of a low Kronecker rank HKT approximation to the class of multidimensional integral operators. In general, given a tolerance $\varepsilon > 0$, we have the bound

$$r = \mathcal{O}\left(\left[\log\left(h^{-1}\right)\log\left(\varepsilon^{-1}\right)\log\left(\log\varepsilon^{-1}\right)\right]^{d-1}\right),$$

where h is the mesh parameter of the finite element discretization. However, for the class of translation-invariant kernels (see [17] and the examples below), we obtain a dimensionally independent bound

$$r = \mathcal{O}\left(\log\left(h^{-1}\right)\log\left(\varepsilon^{-1}\right)\log\left(\log\varepsilon^{-1}\right)\right).$$

4.2. Tucker-HKT approximation. Following Definition 3.2, we introduce the dth order FGT(G) that represents the integral operator $\mathbb{G}$,

$$\mathcal{A} \equiv \mathcal{A}(g) := [a_{\mathbf{m}_1\dots\mathbf{m}_d}] \in \mathbb{R}^{\mathcal{M}_1\times\cdots\times\mathcal{M}_d}$$

with

$$a_{\mathbf{m}_1\dots\mathbf{m}_d} := \int_{\Omega\times\Omega} g(\zeta^{(1)},\dots,\zeta^{(d)})\phi^{\mathbf{i}}(x)\phi^{\mathbf{j}}(y)dxdy, \tag{4.4}$$

where

$$\zeta^{(\ell)} = (x_\ell, y_\ell),\ \mathbf{m}_\ell = (i_\ell, j_\ell) \in \mathcal{M}_\ell \equiv I_\ell \times I_\ell,\ \ell = 1,\dots,d.$$

Assume that the kernel function $g(x,y) \equiv g(\zeta^{(1)},\dots,\zeta^{(d)})$ allows a separable approximation (3.8) via the Sinc-interpolation that converges exponentially in $r = \max_\ell r_\ell$

(see Corollary 3.6). Then the associated rank-$(r_1,\dots,r_d)$ Tucker decomposition in $\boldsymbol{\mathcal{T}}_{\mathbf{r}}$ (cf. Definition 2.1),

$$\mathcal{A}_{(\mathbf{r})} = \sum_{k_1=1}^{r_1} \dots \sum_{k_d=1}^{r_d} b_{k_1\dots k_d} V_{k_1}^{(1)} \otimes \dots \otimes V_{k_d}^{(d)} \in \mathbb{R}^{\mathcal{M}_1\times\dots\times\mathcal{M}_d}, \tag{4.5}$$

is specified by the Kronecker factors $V_{k_\ell}^{(\ell)} \in \mathbb{R}^{\mathcal{M}_\ell}$, explicitly defined by

$$V_{k_\ell}^{(\ell)} = \int \Phi_{k_\ell}^{(\ell)}(\zeta_{\mathbf{i}_\ell}^{(\ell)})\phi_\ell^{i_\ell}(x_\ell)\phi_\ell^{j_\ell}(y_\ell)dx_\ell dy_\ell \in \mathbb{R}^{\mathcal{M}_\ell}, \quad \ell = 1,\dots,d, \quad k_\ell = 1,\dots,r_\ell. \tag{4.6}$$

Let $\mathbf{r} = (r,\dots,r)$. Corollary 3.6 now yields the error estimate

$$\|\mathcal{A}(g) - \mathcal{A}_{(\mathbf{r})}\|_\infty \le Ce^{-\alpha M^\nu} \quad \text{with } r = 2M+1, \tag{4.7}$$

and with constants α, ν from (3.11).

As was already mentioned, (4.7) yields $\max_\ell r_\ell = O(|\log\varepsilon|\delta^{-1})$. In turn, for a class of shift-invariant kernels we get the estimate $\delta^{-1} = O(\log n)$. The numerical complexity of the Tucker decomposition (4.5) is estimated by $drn^2 + r^d$. Applying Lemma 4.2 we conclude that the functions $\Phi_{k_\ell}^{(\ell)}$ are asymptotically smooth; hence the canonical components $V_{k_\ell}^{(\ell)}$ can be approximated by a rank-m $\mathcal{H}$-matrix $\widetilde{V}_{k_\ell}^{(\ell)}$ with an error

$$\|V_{k_\ell}^{(\ell)} - \widetilde{V}_{k_\ell}^{(\ell)}\| \le C\eta^m \quad \text{for some} \quad \eta < 1.$$

The storage cost for the corresponding Tucker-HKT approximation in $\boldsymbol{\mathcal{T}}_{\mathcal{H},\mathbf{r}}$ has the complexity $drnm\log^q n + r^d$, while the complexity of further matrix operations is estimated in Lemma 2.3. Notice that the Tucker-HKT approximation can be applied to more general kernel functions compared with the canonical-HKT representation (the latter is restricted to the class of translation-invariant kernels).

4.3. Examples: Newton, Yukawa, and Helmholtz potentials. Let $x, y \in \mathbb{R}^d$, $p = 2$, and define $\rho = |x-y|^2 = \zeta_1^2 + \dots + \zeta_d^2$ with $\zeta_\ell = x_\ell - y_\ell : \mathbb{R}^2 \to \mathbb{R}$, $\boldsymbol{\zeta} \in \mathbb{R}^d$. The family of functions

$$g(x,y) \equiv g(\boldsymbol{\zeta}) := 1/\rho^\lambda \quad \text{with } \lambda \in \mathbb{R}_{>0}$$

arises in potential theory, in quantum chemistry, and in computational gas dynamics (cf. [23]). The choice $\lambda = 1/2$ corresponds to the classical Newton potential, while $\lambda = -1/2$ refers to the distance function.

Low separation rank decomposition to the multivariate functions $1/\rho$, $1/\sqrt{\rho}$ and to the related Galerkin approximations were discussed in [2, 17, 18, 19, 24], while the kernel function ρ^μ, $\mu \in \mathbb{R}$, was considered in [23].

Let us take a closer look at the FGT(G) for the Newton potential $1/\sqrt{\rho}$ in the hypercube $[-R,R]^d \in \mathbb{R}^d$. As a basic example, we consider piecewise linear finite elements defined by scaling functions $\phi(x) = \psi(x)$ associated with a tensor-product grid with step-size $h > 0$.

LEMMA 4.3 (see [24]). *The FGT(G) for the Newton potential $1/\sqrt{\rho}$ in the hypercube $[-R,R]^d \subset \mathbb{R}^d$ allows a CP approximation with exponential convergence rate (independent of d) as in* (3.11) *with $\nu = 1/2$.*

Proof. We apply the FGT(G)-version of Theorem 3.4. In our case we have $\rho_0(x,y) = (x-y)^2$ $(x,y \in \mathbb{R})$; hence, making use of the Gaussian transform

$$\frac{1}{\sqrt{\rho}} = \frac{2}{\sqrt{\pi}} \int_{\mathbb{R}_+} e^{-\rho\tau^2} d\tau,$$

we obtain $\Psi_{i,j}(\tau) = \Psi_{|i-j|}(\tau) := \int_{\mathbb{R}^2} e^{-\tau^2(x-y)^2} \phi(x)\psi(y+|i-j|h)dxdy$, $\tau \geq 0$. Following [26], we choose the analyticity domain as a sector $\Omega_{\mathcal{G}} := \{w \in \mathbb{C} : |\arg(w)| < \delta\}$ with apex angle $0 < 2\delta < \pi/2$ (here $\mathcal{G} = 1$), and then apply the conformal map $\varphi^{-1} : \Omega_{\mathcal{G}} \to D_\delta$ with $w = \varphi(z) = e^z$, $\varphi^{-1}(w) = \log(w)$ (cf. Theorem 3.4(a)).

To check condition (b′) of Theorem 3.4, first we notice that the transformed integrand

$$f(z) := \exp(z) \prod_{\ell=1}^{d} \Psi_{i_\ell j_\ell}(\phi(z))$$

belongs to the Hardy space $H^1(D_\delta)$. Let $H = (H_1, \ldots, H_d) \in \mathbb{R}^d$ with $H_\ell = |i_\ell - j_\ell|h \leq R$ and let

$$(f * g)(u) = \int_{\mathbb{R}} f(x)g(u-x)dx$$

be the convolution product in $\mathbb{R}^d$, provided that $q(x) = |f| * |g|$ is locally integrable. Now, using the shift property of convolution, $f(\cdot + C) * g(\cdot) = f * g(\cdot + C)$ and applying the Fubini theorem in the form

$$(f * g, \mu)_{L^2} = \int_{\mathbb{R}^d \times \mathbb{R}^d} f(x)g(y)\mu(x+y)dxdy, \quad \mu \in \mathcal{D}(\mathbb{R}^d),$$

we obtain (see [24] for more details)

$$\int_{\mathbb{R}^d \times \mathbb{R}^d} e^{-w^2|x-y|^2} \phi(x)\psi(y-H)dxdy = \int_{\mathbb{R}^d} e^{-w^2|v-H|^2} (\phi * \psi)(v)dv,$$

taking into account that the functions ϕ and ψ have compact support $[-h,h]^d$.

Notice that $\operatorname{supp}(\phi * \psi) = [-2h, 2h]^d$. Now we estimate the constant $N(f, D_\delta)$ by

$$\begin{aligned}
N(f, D_\delta) &= \int_{\partial\Omega_{\mathcal{G}}} |f(w)|\,|dw| \\
&= \int_{\partial\Omega_{\mathcal{G}}} \int_{\mathbb{R}^d \times \mathbb{R}^d} \left| e^{-w^2|x-y|^2} \phi(x)\psi(y-H)dxdy \right| |dw| \\
&= 2 \int_{\mathbb{R}_+} \int_{\mathbb{R}^d} \left| e^{-\zeta^2 \exp(2i\delta)|u-H|^2} (\phi * \psi)(u)du \right| d\zeta \\
&\leq 2 \int_{\mathbb{R}^d} \int_{\mathbb{R}_+} \left| e^{-\zeta^2 \exp(2i\delta)|u-H|^2} \right| d\zeta\, |\phi * \psi|(u)du \\
&= 2 \int_{\mathbb{R}^d} \int_{\mathbb{R}_+} e^{-\zeta^2 cos(2\delta)|u-H|^2} d\zeta\, |\phi * \psi|(u)du \\
&\leq \frac{Cmeas(\operatorname{supp}(\phi * \psi))}{diam(\operatorname{supp}(\phi * \psi))\sqrt{cos(2\delta)}}.
\end{aligned}$$

Finally, we check that condition (c1) is also valid, which completes the proof. ∎

A similar result holds for the Yukawa potential (cf. [24]). Moreover, the above arguments can be also applied to the analysis for the Helmholtz potential in the domain $[-R,R]^d$ for a fixed and not too large R.

4.4. Numerical tests. Numerical tests for the analytic CP approximation to the FGT(C) representation of the Newton and some other potentials are given in [17]. In the following, we present the convergence history for the algebraically optimal rank-(r,r,r) Tucker decomposition to the FGT(C) representations of the target integral operators. We make use of the cell-centred collocation on the $30 \times 30 \times 30$ tensor-product grid on $[0,R]^d$, where $R = 10$ for the first two kernel-functions and with $R = 2\pi$ for the Helmholtz kernel.

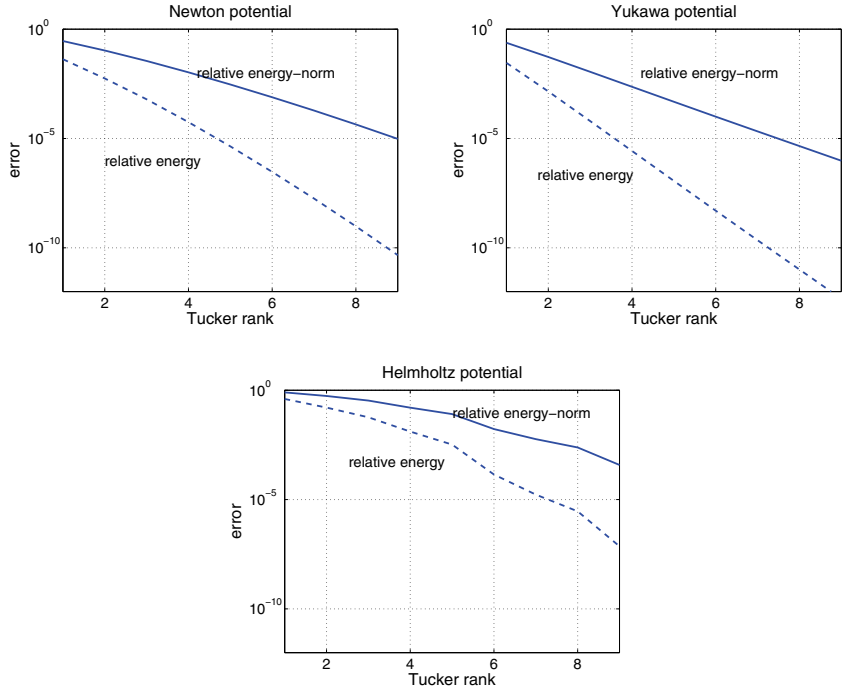


FIG. 4.1. *Rank-$(r,\dots,r)$ Tucker approximation to the Newton (top left), the Yukawa (top right), and the Helmholtz (bottom) potentials for $r = 1,\dots,9$, and with $d = 3$.*

We compute the best rank-(r,r,r) Tucker approximation by the ALS algorithm (implemented in MATLAB) applied to the Lagrange equation for the minimization problem (2.4) with $\mathcal{A} \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$ (cf. [4]). Figure 4.1 indicates the exponential convergence of the best rank-(r,r,r) approximation in r with $r \in [1,9]$ for each of the three potentials considered.

5. Elliptic operator inverse.

5.1. General framework. We consider the elliptic operator $\mathbb{L} : H_0^1(\Omega) \to H^{-1}(\Omega)$ given in the form

$$\mathbb{L} = -\sum_{j=1}^{d} \frac{\partial}{\partial x_j} a_j(x_j) \frac{\partial}{\partial x_j} + \sum_{j=1}^{d} \left[b_j(x_j) \frac{\partial}{\partial x_j} + c_j(x_j) \right], \quad x = (x_1,\dots,x_d) \in \Omega.$$

To derive the tensor-product representation, we employ a finite difference discretization A of $\mathbb{L}$ (e.g., a three-point stencil in each variable) using a uniform tensor-product grid in $\mathbb{R}^d$ with n grid points in each spatial direction. The discretization

matrix has the form $A = \sum_{j=1}^{d} A_j$ with $A, A_j \in \mathbb{R}^{N\times N}$, $N = n^d$, where the matrices A_j are mutually commutable.

Let A be positive definite. The negative fractional power $A^{-\sigma}$ can be represented by

$$A^{-\sigma} = \frac{1}{\Gamma(\sigma)} \int_0^\infty t^{\sigma-1} \mathrm{e}^{-tA} \mathrm{d}t \qquad (\sigma > 0) \tag{5.1}$$

(cf. [11]), provided that the integral exists.

We apply an exponentially convergent quadrature rule (cf. [11, 23]) to represent the integral (5.1) by a sum involving only factorized expressions,

$$A^{-\sigma} \approx \sum_{k=-M}^{M} c_k t_k^{\sigma-1} \prod_{j=1}^{d} \exp(-t_k A_j) \quad (t_k, c_k \in \mathbb{R} \text{ quadrature points and weights}),$$

which leads to the desired HKT representation (2.6). The complexity of the HKT approximation can be estimated by $\mathcal{O}(dn \log^q n)$, where q is some fixed constant independent of d.

Note that with the choice $\mathcal{A} = -\Delta$, the representation (2.6) is of particular interest in the cases $\sigma = 1/2$ (preconditioner of the Laplace–Beltrami operator $(-\Delta)^{1/2}$, and for hypersingular integral operators, e.g., in BEM applications), $\sigma = 1$ (inverse Laplacian), and $\sigma = 2$ (preconditioner for the biharmonic operator).

5.2. Laplace operator inverse. The finite difference "d-dimensional Laplacian" on the uniform $n\times\cdots\times n$ tensor-product grid (subject to homogeneous Dirichlet boundary conditions) takes the form

$$A := V^1 \otimes I \otimes \cdots \times I + I \otimes V^2 \otimes \cdots \otimes I + \cdots + I \otimes I \otimes \cdots \otimes V^d \in \mathbb{R}^{n^d \times n^d}$$

with $V^j, I \in \mathbb{R}^{n\times n}$, where I is the identity matrix and $V^j = \mathrm{tridiag}\{-1, 2, -1\}$, $j = 1, \ldots, d$. We construct the CP approximation in the form

$$A_{(r)} = \sum_{k=-M}^{M} c_k \bigotimes_{\ell=1}^{d} \exp(-t_k V^j) \approx A^{-1} \quad (t_k, c_k \in \mathbb{R}),$$

providing exponential convergence in $r = 2M + 1$. The choice of coefficients t_k, c_k corresponds to the sinc-quadrature rule applied to the integral representation

$$\frac{1}{\rho} = \int_{\mathbb{R}_+} e^{-\rho\tau} d\tau \equiv \int_{\mathbb{R}} e^t e^{-\rho e^t} dt.$$

Figure 5.1 indicates the exponential convergence of the CP approximation $A_{(r)}$ in r in the case of two different quadratures. We calculate (in MATLAB) the Laplace operator inverse on the domain $(0, 1)^d$ with $n = 128$ grid points in each spatial direction. We recall that the memory requirements for our algorithm are estimated by $O(drn)$ (compare with the linear complexity $N = n^d$).

Direct application of Theorem 6.1 in the case of exponential decay (6.4) with $b = 1$ leads to the choice

$$t_k = e^{k\mathfrak{h}}, \quad c_k = \mathfrak{h} t_k, \quad \mathfrak{h} = \pi/\sqrt{M},$$

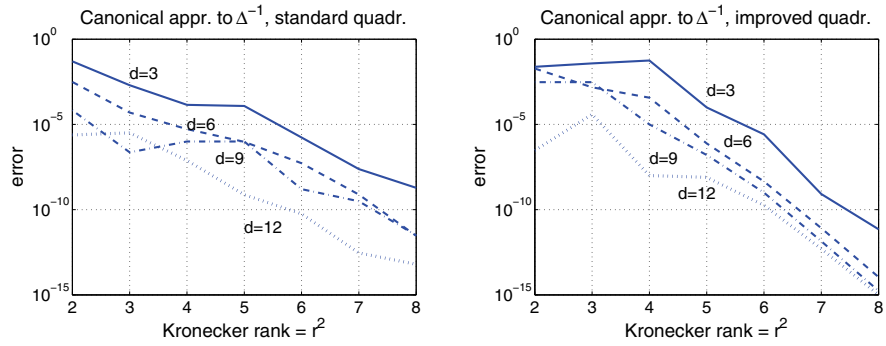


FIG. 5.1. *Rank-r CP approximation to the Laplace operator inverse with* $d = 3, 6, 9, 12$.

providing the convergence rate

$$\left\|A^{-1} - A_{(r)}\right\| \le Ce^{-\pi\sqrt{M}}.$$

This case corresponds to the left part in Figure 5.1. Using a special substitution of variables (see [17] for more details) we obtain an integral with hyperexponential decay as in (6.5),

$$\frac{1}{\rho} = \int_{\mathbb{R}} f_2(w)\mathrm{d}w \quad \text{with } f_2(w) = \frac{\cosh(w)}{1+\mathrm{e}^{-\sinh(w)}}\mathrm{e}^{-\rho\log(1+\mathrm{e}^{\sinh(w)})}. \tag{5.2}$$

Substitution of $\mathfrak{h} = C_0 \log M/M$ into (6.2) leads to the approximation error estimate

$$\left\|A^{-1} - A_{(r)}\right\| \le Ce^{-sM/\log(M)}.$$

This case corresponds to the right-hand part in Figure 5.1.

6. Appendix: Sinc-approximation. Following the standard tools in the sinc methods (cf. [26]), we introduce the Hardy space $H^1(D_\delta)$ as the set of all complex-valued functions f, which are analytic in the strip

$$D_\delta := \{z \in \mathbb{C} : |\Im\mathfrak{m}\, z| < \delta\},$$

such that

$$N(f, D_\delta) := \int_{\partial D_\delta} |f(z)|\,|\mathrm{d}z| = \int_{\mathbb{R}} \left(|f(x+i\delta)| + |f(x-i\delta)|\right)\mathrm{d}x < \infty.$$

Let

$$S(k,\mathfrak{h})(x) = \frac{\sin\left[\pi(x-k\mathfrak{h})/\mathfrak{h}\right]}{\pi(x-k\mathfrak{h})/\mathfrak{h}} \equiv \mathrm{sinc}\left(\frac{x}{\mathfrak{h}} - k\right) \qquad (k \in \mathbb{Z},\ \mathfrak{h} > 0,\ x \in \mathbb{R})$$

be the kth Sinc-function with step-size $\mathfrak{h}$, evaluated at x, where the Sinc-function is defined by

$$\mathrm{sinc}(z) = \frac{\sin(\pi z)}{\pi z}, \quad z \in \mathbb{C}. \tag{6.1}$$

Given $f \in H^1(D_\delta)$, $\mathfrak{h} > 0$, and $M \in \mathbb{N}_0$, the corresponding sinc-quadrature read as

$$T_M(f,\mathfrak{h}) := \mathfrak{h} \sum_{k=-M}^{M} f(k\mathfrak{h}) \approx \int_{\mathbb{R}} f(\xi)\mathrm{d}\xi. \tag{6.2}$$

The classical Sinc interpolant (cardinal series representation) reads as

$$C_M(f,\mathfrak{h}) = \sum_{\nu=-M}^{M} S(\nu,\mathfrak{h}) f(\nu\mathfrak{h}) \approx f. \tag{6.3}$$

THEOREM 6.1. *Let $f \in H^1(D_\delta)$, $\mathfrak{h} > 0$, and $M \in \mathbb{N}_0$ be given. If*

$$|f(\xi)| \le C\exp(-b|\xi|) \qquad \text{for all } \xi \in \mathbb{R} \text{ with } b, C > 0, \tag{6.4}$$

then the quadrature error satisfies

$$\left| \int_{\mathbb{R}} f(\xi)\mathrm{d}\xi - T_M(f,\mathfrak{h}) \right| \le C e^{-\sqrt{2\pi\delta b M}} \quad \text{with } \mathfrak{h} = \sqrt{2\pi\delta/bM},$$

and with a positive constant C depending only on f, δ, b (cf. [26]). *If f possesses the hyperexponential decay*

$$|f(\xi)| \le C\exp(-b\mathrm{e}^{a|\xi|}) \qquad \text{for all } \xi \in \mathbb{R} \quad \text{with } a, b, C > 0, \tag{6.5}$$

then the choice $\mathfrak{h} = \log(\frac{2\pi a M}{b})/(aM)$ leads to (cf. [10])

$$\left| \int_{\mathbb{R}} f(\xi)\mathrm{d}\xi - T_M(f,\mathfrak{h}) \right| \le C\, N(f, D_\delta)\, \mathrm{e}^{-2\pi\delta a M/\log(2\pi a M/b)}.$$

If (6.4) *holds, then the interpolation error satisfies* (cf. [26])

$$\|f - C_M(f,\mathfrak{h})\|_\infty \le C M^{1/2} e^{-\sqrt{\pi\delta b M}} \quad \text{with } \mathfrak{h} = \sqrt{\pi\delta/bM}.$$

Assuming the hyperexponential decay of f, we obtain (cf. [10])

$$\|f - C_M(f,\mathfrak{h})\|_\infty \le C\,\frac{N(f,D_\delta)}{2\pi\delta}\mathrm{e}^{-\pi\delta a M/\log(\pi a M/b)} \quad \text{with } \mathfrak{h} = \log\left(\frac{\pi a M}{b}\right)/(aM).$$

Note that $2M+1$ is the number of quadrature/interpolation points. If f is an even function, the number of quadrature/interpolation points reduces to $M+1$.

The Sinc-interpolation method can be extended to the multidimensional case. Let $g_\ell(\cdot) : I_\ell \to \mathbb{R}$ be a univariate parameter-dependent function, which is the restriction of a multivariate function g onto I_ℓ with fixed remaining variables. Suppose that $g_\ell(\cdot)$ satisfies all the regularity and decay conditions above, uniformly in $\ell = 1, \dots, d$. It is shown in [17] that the *tensor-product Sinc-interpolation* $\mathbf{C}_M$ with respect to d variables,

$$\mathbf{C}_M g := C_M^{(1)} \dots C_M^{(d)} g,$$

provides the exponential error estimate

$$|g(\zeta) - \mathbf{C}_M(g,\mathfrak{h})(\zeta)| \le \frac{C\Lambda_M^d}{2\pi\delta} \max_{\ell=1,\dots,d} N(g_\ell(\cdot), D_\delta)\, \mathrm{e}^{\frac{-\pi\delta M}{\log M}}$$

with the Lebesgue constant $\Lambda_M = O(\log M)$, and where $C_M^{(\ell)} g = C_M^{(\ell)}(g,\mathfrak{h})$ denotes the univariate Sinc-interpolation from (6.3) applied to the variable $\zeta_\ell \in I_\ell$.

REFERENCES

[1] G. Beylkin and M. J. Mohlenkamp, *Numerical operator calculus in higher dimensions*, Proc. Natl. Acad. Sci., 99 (2002), pp. 10246–10251.

[2] G. Braess and W. Hackbusch, *Approximation of* $1/x$ *by exponential sums in* $[1, \infty)$, IMA J. Numer. Anal., 25 (2005), pp. 685–697.

[3] J. D. Carrol and J. Chang, *Analysis of individual differences in multidimensional scaling via an* N*-way generalization of "Eckart–Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[4] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-*1 *and rank-*$(R_1, R_2, \ldots, R_N)$ *approximation of higher-order tensors,* SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[5] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition,* SIAM J. Matrix Anal. Appl., 26 (2004), pp. 295–327.

[6] H.-J. Flad, W. Hackbusch, B. N. Khoromskij, and R. Schneider, *Concept of data-sparse tensor-product approximation in many-particle models*, in preparation.

[7] H.-J. Flad, W. Hackbusch, D. Kolb, and R. Schneider, *Wavelet approximation of correlated wavefunctions.* I. *Basics*, J. Chem. Phys., 116 (2002), pp. 9641–9657.

[8] I. P. Gavrilyuk, W. Hackbusch, and B. N. Khoromskij, $\mathcal{H}$*-matrix approximation for the operator exponential with applications*, Numer. Math., 92 (2002), pp. 83–111.

[9] I. P. Gavrilyuk, W. Hackbusch, and B. N. Khoromskij, *Data-sparse approximation to operator-valued functions of elliptic operators*, Math. Comp., 73 (2004), pp. 1297–1324.

[10] I. P. Gavrilyuk, W. Hackbusch, and B. N. Khoromskij, *Data-sparse approximation to a class of operator-valued functions*, Math. Comp., 74 (2005), pp. 681–708.

[11] I. P. Gavrilyuk, W. Hackbusch, and B. N. Khoromskij, *Tensor-product approximation to elliptic and parabolic solution operators in higher dimensions*, Computing, 74 (2005), pp. 131–157.

[12] L. Grasedyck and W. Hackbusch, *Construction and arithmetics of* $\mathcal{H}$*-matrices*, Computing, 70 (2003), pp. 295–334.

[13] L. Grasedyck, W. Hackbusch, and B. N. Khoromskij, *Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices*, Computing, 70 (2003), pp. 121–165.

[14] W. Hackbusch, *A sparse matrix arithmetic based on* $\mathcal{H}$*-matrices. Part* I: *Introduction to* $\mathcal{H}$*-matrices*, Computing, 62 (1999), pp. 89–108.

[15] W. Hackbusch and B. N. Khoromskij, *A sparse* $\mathcal{H}$*-matrix arithmetic. Part* II: *Application to multi-dimensional problems*, Computing, 64 (2000), pp. 21–47.

[16] W. Hackbusch and B. N. Khoromskij, *A sparse* $\mathcal{H}$*-matrix arithmetic: General complexity estimates*, J. Comput. Appl. Math., 125 (2000), pp. 479–501.

[17] W. Hackbusch and B. N. Khoromskij, *Low-rank Kronecker product approximation to multi-dimensional nonlocal operators. Part* I. *Separable approximation of multi-variate functions*, Computing, 76 (2006), pp. 177–202.

[18] W. Hackbusch and B. N. Khoromskij, *Low-rank Kronecker product approximation to multi-dimensional nonlocal operators. Part* II. *HKT representations of certain operators*, Computing, 76 (2006), pp. 203–225.

[19] W. Hackbusch, B. N. Khoromskij, and E. Tyrtyshnikov, *Hierarchical Kronecker tensor-product approximation*, J. Numer. Math., 13 (2005), pp. 119–156.

[20] W. Hackbusch, B. N. Khoromskij, and E. E. Tyrtyshnikov, *Approximate Iterations for Structured Matrices*, Preprint 112, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, 2005.

[21] R. Harshman, *Foundation of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis,* UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[22] B. N. Khoromskij, *An Introduction to Structured Tensor-Product Representation of Discrete Nonlocal Operators*, Lecture Notes 27, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, 2005.

[23] B. N. Khoromskij, *Structured data-sparse approximation to high order tensors arising from the deterministic Boltzmann equation*, Math. Comp., 76 (2007), pp. 1291–1315.

[24] B. N. Khoromskij, *Structured rank-*$(r_1, \ldots, r_d)$ *decomposition of function-related tensors in* $\mathbb{R}^d$, Comput. Methods Appl. Math., 6 (2006), pp. 194–220.

[25] T. G. Kolda, *Orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 243–255.

[26] F. Stenger, *Numerical Methods Based on Sinc and Analytic Functions,* Springer-Verlag, New York, 1993.

[27] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[28] E. E. Tyrtyshnikov, *Tensor approximations of matrices generated by asymptotically smooth functions,* Mat. Sb., 194 (2003), pp. 147–160 (in Russian); translation in Sb. Math., 194 (2003), pp. 941–954.

[29] N. Yarvin and V. Rokhlin, *Generalized Gaussian quadratures and singular value decompositions of integral operators,* SIAM J. Sci. Comput., 20 (1998), pp. 699–718.

[30] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 3, pp. 1254–1279

SYMMETRIC TENSORS AND SYMMETRIC TENSOR RANK*

PIERRE COMON†, GENE GOLUB‡, LEK-HENG LIM‡, AND BERNARD MOURRAIN§

Abstract. A symmetric tensor is a higher order generalization of a symmetric matrix. In this paper, we study various properties of symmetric tensors in relation to a decomposition into a symmetric sum of outer product of vectors. A rank-1 order-k tensor is the outer product of k nonzero vectors. Any symmetric tensor can be decomposed into a linear combination of rank-1 tensors, each of which is symmetric or not. The *rank* of a symmetric tensor is the minimal number of rank-1 tensors that is necessary to reconstruct it. The *symmetric rank* is obtained when the constituting rank-1 tensors are imposed to be themselves symmetric. It is shown that rank and symmetric rank are equal in a number of cases and that they always exist in an algebraically closed field. We will discuss the notion of the generic symmetric rank, which, due to the work of Alexander and Hirschowitz [*J. Algebraic Geom.*, 4 (1995), pp. 201–222], is now known for any values of dimension and order. We will also show that the set of symmetric tensors of symmetric rank at most r is not closed unless $r = 1$.

Key words. tensors, multiway arrays, outer product decomposition, symmetric outer product decomposition, CANDECOMP, PARAFAC, tensor rank, symmetric rank, symmetric tensor rank, generic symmetric rank, maximal symmetric rank, quantics

AMS subject classifications. 15A03, 15A21, 15A72, 15A69, 15A18

DOI. 10.1137/060661569

1. Introduction.

We will be interested in the decomposition of a symmetric tensor into a minimal linear combination of symmetric outer products of vectors (i.e., of the form $\mathbf{v} \otimes \mathbf{v} \otimes \cdots \otimes \mathbf{v}$). We will see that a decomposition of the form

$$A = \sum_{i=1}^{r} \lambda_i \mathbf{v}_i \otimes \mathbf{v}_i \otimes \cdots \otimes \mathbf{v}_i \tag{1.1}$$

always exists for any symmetric tensor A (over any field). One may regard this as a generalization of the eigenvalue decomposition for symmetric matrices to higher order symmetric tensors. In particular, this will allow us to define a notion of *symmetric tensor rank* (as the minimal r over all such decompositions) that reduces to the matrix rank for order-2 symmetric tensors.

We will call (1.1) the *symmetric outer product decomposition* of the symmetric tensor A and we will establish its existence in Proposition 4.2. This is often abbreviated as CANDECOMP in signal processing. The decomposition of a tensor into an (asymmetric) outer product of vectors and the corresponding notion of tensor rank was first introduced and studied by Frank L. Hitchcock in 1927 [29, 30]. This same decomposition was rediscovered in the 1970s by psychometricians in their attempts to define

*Received by the editors June 1, 2006; accepted for publication (in revised form) by L. De Lathauwer February 1, 2008; published electronically September 25, 2008.
http://www.siam.org/journals/simax/30-3/66156.html

†Laboratoire I3S, CNRS, and the University of Nice, 06093 Sophia-Antipolis, France (p.comon@ieee.org). This author's research was partially supported by contract ANR-06-BLAN-0074.

‡Department of Computer Science and the Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 (golub@cs.stanford.edu, lekheng@cs.stanford.edu). The second author's research was partially supported by NSF grant CCF 0430617. The third author's research was partially supported by NSF grant DMS 0101364 and by the Gerald J. Lieberman Fellowship from Stanford University.

§Projet GALAAD, INRIA, 06902 Sophia-Antipolis, France (mourrain@sophia.inria.fr). This author's research was partially supported by contract ANR-06-BLAN-0074 "DECOTES."

data analytic models that generalize factor analysis to multiway data [60]. The name CANDECOMP, for canonical decompositions, was used by Carroll and Chang [11] and the name PARAFAC, for parallel factor analysis, was used by Harshman [28] for their respective models.

The symmetric outer product decomposition is particularly important in the process of *blind identification of underdetermined mixtures* (UDM), i.e., linear mixtures with more inputs than observable outputs. See [14, 17, 20, 50, 51] and references therein for a list of other application areas, including speech, mobile communications, machine learning, factor analysis of k-way arrays, biomedical engineering, psychometrics, and chemometrics.

Despite a growing interest in the symmetric decomposition of symmetric tensors, this topic has not been adequately addressed in the general literature, and even less so in the engineering literature. For several years, the alternating least squares algorithm has been used to fit data arrays to a multilinear model [36, 51]. Yet the minimization of this matching error is an ill-posed problem in general, since the set of symmetric tensors of symmetric rank not more than r is not closed, unless $r = 1$ (see sections 6 and 8)—a fact that parallels the ill-posedness discussed in [21]. The focus of this paper is mainly on symmetric tensors. The asymmetric case will be addressed in a companion paper, and will use similar tools borrowed from algebraic geometry.

Symmetric tensors form a singularly important class of tensors. Examples where these arise are higher order derivatives of smooth functions [40] and moments and cumulants of random vectors [44]. The decomposition of such symmetric tensors into simpler ones, as in the symmetric outer product decomposition, plays an important role in *independent component analysis* [14] and constitutes a problem of interest in its own right. On the other hand, the asymmetric version of the outer product decomposition defined in (4.1) is central to *multiway factor analysis* [51].

In sections 2 and 3, we discuss some classical results in multilinear algebra [5, 26, 39, 42, 45, 64] and algebraic geometry [27, 65]. While these background materials are well known to many pure mathematicians, we found that practitioners and applied mathematicians (in signal processing, neuroimaging, numerical analysis, optimization, etc.)—for whom this paper is intended—are often unaware of these classical results. For instance, some do not realize that the classical definition of a symmetric tensor given in Definition 3.2 is equivalent to the requirement that the coordinate array representing the tensor be invariant under all permutations of indices, as in Definition 3.1. Many authors have persistently mislabeled the latter a "*supersymmetric* tensor" (cf. [10, 34, 46]). In fact, we have found that even the classical definition of a symmetric tensor is not as well known as it should be. We see this as an indication of the need to inform our target readership. It is our hope that the background materials presented in sections 2 and 3 will serve such a purpose.

Our contributions begin in section 4, where the notions of maximal and generic rank are analyzed. The concepts of *symmetry* and *genericity* are recalled in sections 3 and 4, respectively. The distinction between symmetric rank and rank is made in section 4, and it is shown in section 5 that they must be equal in specific cases. It is also pointed out in section 6 that the generic rank always exists in an algebraically closed field and that it is not maximal except in the binary case. More precisely, the sequence of sets of symmetric tensors of symmetric rank r increases with r (in the sense of inclusion) up to the generic symmetric rank and *decreases* thereafter. In addition, the set of symmetric tensors of symmetric rank at most r and order $d > 2$ is closed only for $r = 1$ and $r = R_S$, the maximal symmetric rank. Values of the generic symmetric rank and the uniqueness of the symmetric outer product decomposition

are addressed in section 7. In section 8, we give several examples of sequences of symmetric tensors converging to limits having strictly higher symmetric ranks. We also give an explicit example of a symmetric tensor whose values of symmetric rank over $\mathbb{R}$ and over $\mathbb{C}$ are different.

In this paper, we restrict our attention mostly to decompositions over the complex field. A corresponding study over the real field will require techniques rather different from those introduced here, as we will elaborate in section 8.2.

2. Arrays and tensors. A *k-way array* of complex numbers will be written in the form $A = [\![a_{j_1\cdots j_k}]\!]_{j_1,\ldots,j_k=1}^{n_1,\ldots,n_k}$, where $a_{j_1\cdots j_k} \in \mathbb{C}$ is the $(j_1,\ldots,j_k)$-entry of the array. This is sometimes also called a k-dimensional *hypermatrix*. We denote the set of all such arrays by $\mathbb{C}^{n_1\times\cdots\times n_k}$, which is evidently a complex vector space of dimension $n_1\cdots n_k$ with respect to entrywise addition and scalar multiplication. When there is no confusion, we will leave out the range of the indices and simply write $A = [\![a_{j_1\cdots j_k}]\!] \in \mathbb{C}^{n_1\times\cdots\times n_k}$.

Unless noted otherwise, arrays with at least two indices will be denoted in uppercase; vectors are one-way arrays and will be denoted in bold lowercase. For our purpose, only a few notations related to arrays [14, 20] are necessary.

The *outer product* (or *Segre outer product*) of k vectors $\mathbf{u} \in \mathbb{C}^{n_1}, \mathbf{v} \in \mathbb{C}^{n_2}, \ldots, \mathbf{z} \in \mathbb{C}^{n_k}$ is defined as

$$\mathbf{u}\otimes\mathbf{v}\otimes\cdots\otimes\mathbf{z} := [\![u_{j_1}v_{j_2}\cdots z_{j_k}]\!]_{j_1,j_2,\ldots,j_k=1}^{n_1,n_2,\ldots,n_k}.$$

More generally, the outer product of two arrays A and B, respectively, of orders k and ℓ is an array of order $k+\ell$, $C = A\otimes B$ with entries

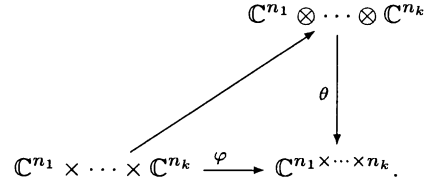
$$c_{i_1\cdots i_k j_1\cdots j_\ell} := a_{i_1\cdots i_k} b_{j_1\cdots j_\ell}.$$

For example, the outer product of two vectors, $\mathbf{u}\otimes\mathbf{v}$, is a matrix. The outer product of three vectors, or of a matrix with a vector, is a 3-way array.

How is an array related to a tensor? Recall that a tensor is simply an element in the tensor product of vector spaces [5, 26, 39, 42, 45, 64]. One may easily check that the so-called Segre map

$$\begin{aligned}\varphi : \mathbb{C}^{n_1}\times\cdots\times\mathbb{C}^{n_k} &\to \mathbb{C}^{n_1\times\cdots\times n_k},\\ (\mathbf{u},\ldots,\mathbf{z}) &\mapsto \mathbf{u}\otimes\cdots\otimes\mathbf{z}\end{aligned}$$

is multilinear. By the universal property of the tensor product [5, 26, 39, 42, 45, 64], there exists a linear map θ



Since $\dim(\mathbb{C}^{n_1}\otimes\cdots\otimes\mathbb{C}^{n_k}) = \dim(\mathbb{C}^{n_1\times\cdots\times n_k})$, θ is an isomorphism of the vector spaces $\mathbb{C}^{n_1}\otimes\cdots\otimes\mathbb{C}^{n_k}$ and $\mathbb{C}^{n_1\times\cdots\times n_k}$. Consider the canonical basis of $\mathbb{C}^{n_1}\otimes\cdots\otimes\mathbb{C}^{n_k}$,

$$\{\mathbf{e}_{j_1}^{(1)}\otimes\cdots\otimes\mathbf{e}_{j_k}^{(k)} \mid 1\le j_1\le n_1,\ldots,1\le j_k\le n_k\},$$

where $\{\mathbf{e}_1^{(\ell)}, \dots, \mathbf{e}_{n_\ell}^{(\ell)}\}$ denotes the canonical basis in $\mathbb{C}^{n_\ell}$, $\ell = 1, \dots, k$. Then θ may be described by

$$\theta\Big(\sum\nolimits_{j_1,\dots,j_k=1}^{n_1,\dots,n_k} a_{j_1,\dots,j_k} \mathbf{e}_{j_1}^{(1)} \otimes \cdots \otimes \mathbf{e}_{j_k}^{(k)}\Big) = [\![a_{j_1\cdots j_k}]\!]_{j_1,\dots,j_k=1}^{n_1,\dots,n_k}.$$

So an order-k tensor in $\mathbb{C}^{n_1} \otimes \cdots \otimes \mathbb{C}^{n_k}$ and a k-way array in $\mathbb{C}^{n_1 \times \cdots \times n_k}$ that represents the tensor with respect to a basis may be regarded as synonymous (up to, of course, the choice of basis). We will illustrate how the k-array representation of an order-k tensor is affected by a change-of-basis. Let $A = [\![a_{ijk}]\!] \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ and let L, M, and N be three matrices of size $r_1 \times n_1$, $r_2 \times n_2$, and $r_3 \times n_3$, respectively. Then the tensor A may be transformed by the multilinear map (L, M, N) into a tensor $A' = [\![a'_{pqr}]\!] \in \mathbb{C}^{r_1 \times r_2 \times r_3}$ defined by

$$a'_{pqr} = \sum\nolimits_{i,j,k} l_{pi} m_{qj} n_{rk} a_{ijk}. \tag{2.1}$$

When $r_i = n_i$ and L, M, N are nonsingular matrices, the above multilinear map may be thought of as a change-of-bases (refer to [21] for further discussions). We will call this map a *multilinear transform* of A.

In addition to the outer product, we also have an *inner product* or *contraction product* of two arrays. The mode-p inner product between two arrays A, B having the same pth dimension is denoted $A \bullet_p B$ and is obtained by summing over the pth index. More precisely, if A and B are of orders k and ℓ, respectively, this yields for $p = 1$ the array $C = A \bullet_1 B$ of order $k + \ell - 2$:

$$c_{i_2 \cdots i_k j_2 \cdots j_\ell} = \sum\nolimits_{\alpha} a_{\alpha i_2 \cdots i_k} b_{\alpha j_2 \cdots j_\ell}.$$

Note that some authors [20, 24, 60] denoted this contraction product as $A \times_p B$ or $\langle A, B \rangle_p$. By convention, when the contraction is between a tensor and a matrix, it is convenient to assume that the summation is always done on the second matrix index. For instance, the multilinear transform in (2.1) may be expressed as $A' = A \bullet_1 L \bullet_2 M \bullet_3 N$. An alternative notation for (2.1) from the theory of group actions is $A' = (L, M, N) \cdot A$, which may be viewed as multiplying A on "three sides" by the matrices L, M, and N [21, 32].

3. Symmetric arrays and symmetric tensors. We shall say that a k-way array is *cubical* if all its k dimensions are identical, i.e., $n_1 = \cdots = n_k = n$. A cubical array will be called *symmetric* if its entries do not change under any permutation of its k indices. Formally, if $\mathfrak{S}_k$ denotes the symmetric group of permutations on $\{1, \dots, k\}$, then we have the following definition.

DEFINITION 3.1. *A k-way array $[\![a_{j_1 \cdots j_k}]\!] \in \mathbb{C}^{n \times \cdots \times n}$ is called* symmetric *if*

$$a_{i_{\sigma(1)} \cdots i_{\sigma(k)}} = a_{i_1 \cdots i_k}, \qquad i_1, \dots, i_k \in \{1, \dots, n\},$$

for all permutations $\sigma \in \mathfrak{S}_k$.

For example, a 3-way array $[\![a_{ijk}]\!] \in \mathbb{C}^{n \times n \times n}$ is symmetric if

$$a_{ijk} = a_{ikj} = a_{jik} = a_{jki} = a_{kij} = a_{kji}$$

for all $i, j, k \in \{1, \dots, n\}$.

Such arrays have been improperly labeled "supersymmetric" tensors (cf. [10, 34, 46] among others); this terminology should be avoided since it refers to an entirely

different class of tensors [7]. The word "supersymmetric" has *always* been used in both mathematics and physics [25, 61, 63] to describe objects with a $\mathbb{Z}_2$-grading, and so using it in the sense of [10, 34, 46] is both inconsistent and confusing. (The correct usage will be one in the sense of [7].) In fact, we will show in Proposition 3.7 that there is no difference between Definition 3.1 and the usual definition of a symmetric tensor in mathematics [5, 26, 39, 42, 45, 64]. In other words, the prefix "super" in "supersymmetric tensor," when used in the sense of [10, 34, 46], is superfluous.

We will write $\mathsf{T}^k(\mathbb{C}^n) := \mathbb{C}^n \otimes \cdots \otimes \mathbb{C}^n$ (k copies), the set of all order-k dimension-n cubical tensors. We define a group action $\mathfrak{S}_k$ on $\mathsf{T}^k(\mathbb{C}^n)$ as follows. For any $\sigma \in \mathfrak{S}_k$ and $\mathbf{x}_{i_1} \otimes \cdots \otimes \mathbf{x}_{i_k} \in \mathsf{T}^k(\mathbb{C}^n)$, we let

$$\sigma(\mathbf{x}_{i_1} \otimes \cdots \otimes \mathbf{x}_{i_k}) := \mathbf{x}_{i_{\sigma(1)}} \otimes \cdots \otimes \mathbf{x}_{i_{\sigma(k)}}$$

and extend this linearly to all of $\mathsf{T}^k(\mathbb{C}^n)$. Thus each $\sigma \in \mathfrak{S}_k$ defines a linear operator $\sigma : \mathsf{T}^k(\mathbb{C}^n) \to \mathsf{T}^k(\mathbb{C}^n)$. The standard definition of a symmetric tensor in mathematics [5, 26, 39, 42, 45, 64] looks somewhat different from Definition 3.1 and is given as follows.

DEFINITION 3.2. *An order-k tensor $A \in \mathsf{T}^k(\mathbb{C}^n)$ is* symmetric *if*

$$\sigma(A) = A \tag{3.1}$$

for all permutations $\sigma \in \mathfrak{S}_k$. The set of symmetric tensors in $\mathsf{T}^k(\mathbb{C}^n)$ will be denoted by $\mathsf{S}^k(\mathbb{C}^n)$.

Let $S : \mathsf{T}^k(\mathbb{C}^n) \to \mathsf{T}^k(\mathbb{C}^n)$ be the linear operator defined by

$$S := \frac{1}{k!} \sum\nolimits_{\sigma \in \mathfrak{S}_k} \sigma.$$

Note that given any $\sigma \in \mathfrak{S}_k$,

$$\sigma \circ S = S \circ \sigma = S.$$

Here $\circ$ denotes the composition of the linear operators σ and S.

PROPOSITION 3.3. *An order-k tensor $A \in \mathsf{T}^k(\mathbb{C}^n)$ is symmetric if and only if*

$$S(A) := \frac{1}{k!} \sum\nolimits_{\sigma \in \mathfrak{S}_k} \sigma(A) = A.$$

Proof. Clearly, if A is symmetric, then

$$S(A) = \frac{1}{k!} \sum\nolimits_{\sigma \in \mathfrak{S}_k} \sigma(A) = \frac{1}{k!} \sum\nolimits_{\sigma \in \mathfrak{S}_k} A = A.$$

Conversely, if $S(A) = A$, then

$$\sigma(A) = \sigma(S(A)) = \sigma \circ S(A) = S(A) = A$$

for all $\sigma \in \mathfrak{S}_k$; and so A is symmetric. ∎

In other words, a symmetric tensor is an eigenvector of the linear operator S with eigenvalue 1. $\mathsf{S}^k(\mathbb{C}^n)$ is the 1-eigenspace of $S : \mathsf{T}^k(\mathbb{C}^n) \to \mathsf{T}^k(\mathbb{C}^n)$. Proposition 3.3 implies that $\mathsf{S}^k(\mathbb{C}^n) = S(\mathsf{T}^k(\mathbb{C}^n))$ and it is also easy to see that S is a projection of $\mathsf{T}^k(\mathbb{C}^n)$ onto the subspace $\mathsf{S}^k(\mathbb{C}^n)$, i.e., $S^2 = S$.

3.1. Equivalence with homogeneous polynomials. We adopt the following standard shorthand. For any $\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_k} \in \mathbb{C}^n$ with $i_1, \dots, i_k \in \{1, \dots, n\}$, we write

$$\mathbf{e}_{i_1} \cdots \mathbf{e}_{i_k} := S(\mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_k}) = \frac{1}{k!} \sum\nolimits_{\sigma \in \mathfrak{S}_k} \mathbf{e}_{i_{\sigma(1)}} \otimes \cdots \otimes \mathbf{e}_{i_{\sigma(k)}}. \tag{3.2}$$

Then since $S\sigma = S$, the term $\mathbf{e}_{i_1} \cdots \mathbf{e}_{i_k}$ depends only on the number of times each $\mathbf{e}_i$ enters this product and we may write

$$\mathbf{e}_{i_1} \cdots \mathbf{e}_{i_k} = \mathbf{e}_1^{p_1} \cdots \mathbf{e}_n^{p_n}, \tag{3.3}$$

where p_i is the multiplicity (which may be 0) of occurrence of $\mathbf{e}_i$ in $\mathbf{e}_{i_1} \cdots \mathbf{e}_{i_k}$. Note that $p_1, \dots, p_n$ are nonnegative integers satisfying $p_1 + \cdots + p_n = k$.

PROPOSITION 3.4. *Let $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ be a basis of $\mathbb{C}^n$. Then*

$$\{S(\mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_k}) \mid 1 \le i_1 \le \cdots \le i_k \le n\}$$

or, explicitly,

$$\Big\{ \frac{1}{k!} \sum\nolimits_{\sigma \in \mathfrak{S}_k} \mathbf{e}_{i_{\sigma(1)}} \otimes \cdots \otimes \mathbf{e}_{i_{\sigma(k)}} \;\Big|\; 1 \le i_1 \le \cdots \le i_k \le n \Big\}$$

is a basis of $\mathsf{S}^k(\mathbb{C}^n)$. Furthermore,

$$\dim_{\mathbb{C}} \mathsf{S}^k(\mathbb{C}^n) = \binom{n+k-1}{k}.$$

Proof. Since $\mathcal{B} = \{\mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_k} \mid 1 \le i_1 \le n, \dots, 1 \le i_k \le n\}$ is a basis for $\mathsf{T}^k(\mathbb{C}^n)$ and since S maps $\mathsf{T}^k(\mathbb{C}^n)$ onto $\mathsf{S}^k(\mathbb{C}^n)$, the set

$$S(\mathcal{B}) = \{\mathbf{e}_{i_1} \cdots \mathbf{e}_{i_k} \mid 1 \le i_1 \le \cdots \le i_k \le n\} = \{\mathbf{e}_1^{p_1} \cdots \mathbf{e}_n^{p_n} \mid p_1 + \cdots + p_n = k\}$$

spans $\mathsf{S}^k(\mathbb{C}^n)$. Vectors in $S(\mathcal{B})$ are linearly independent: if $(p_1, \dots, p_n) \neq (q_1, \dots, q_n)$, then the tensors $\mathbf{e}_1^{p_1} \cdots \mathbf{e}_n^{p_n}$ and $\mathbf{e}_1^{q_1} \cdots \mathbf{e}_n^{q_n}$ are, respectively, linear combinations of two nonintersecting subsets of basis elements of $\mathsf{T}^k(\mathbb{C}^n)$. The cardinality of $S(\mathcal{B})$ is precisely the number of partitions of k into a sum of n nonnegative integers, i.e., $\binom{n+k-1}{k}$. ∎

If we regard $\mathbf{e}_j$ in (3.3) as variables (i.e., indeterminates), then every symmetric tensor of order k and dimension n may be uniquely associated with a *homogeneous polynomial* of degree k in n variables. Recall that these are just polynomials in n variables whose constituting monomials all have the same total degree k. Homogeneous polynomials are also called *quantics* and those of degrees 1, 2, and 3 are often called *linear forms*, *quadratic forms*, and *cubic forms* (or just *cubics*), respectively. From now on, we will use more standard notation for the variables—x_j instead of $\mathbf{e}_j$. So the monomial on the right-hand side of (3.3) now becomes $x_1^{p_1} \cdots x_n^{p_n}$. To further simplify this notation, we will adopt the following standard *multi-index notation*:

$$\mathbf{x}^{\boldsymbol{p}} := \prod\nolimits_{k=1}^{n} x_k^{p_k} \quad \text{and} \quad |\boldsymbol{p}| := \sum\nolimits_{k=1}^{n} p_k,$$

where $\boldsymbol{p}$ denotes a k-vector of nonnegative integers. We will also write $\mathbb{C}[x_1, \dots, x_n]_k$ for the set of homogeneous polynomials of degree k in n variables (again a standard notation). Then any symmetric tensor $[\![a_{j_1 \cdots j_k}]\!] = [\![a_{\boldsymbol{j}}]\!] \in \mathsf{S}^k(\mathbb{C}^n)$ can be associated with a unique homogeneous polynomial $F \in \mathbb{C}[x_1, \dots, x_n]_k$ via the expression

$$F(\mathbf{x}) = \sum\nolimits_{\boldsymbol{j}} a_{\boldsymbol{j}} \mathbf{x}^{\boldsymbol{p}(\boldsymbol{j})}, \tag{3.4}$$

where for every $\boldsymbol{j} = (j_1, \ldots, j_k)$, one associates bijectively the nonnegative integer vector $\boldsymbol{p}(\boldsymbol{j}) = (p_1(\boldsymbol{j}), \ldots, p_n(\boldsymbol{j}))$ with $p_j(\boldsymbol{j})$ counting the number of times index j appears in $\boldsymbol{j}$ [16, 14]. We have in particular $|\boldsymbol{p}(\boldsymbol{j})| = k$. The converse is true as well, and the correspondence between symmetric tensors and homogeneous polynomials is obviously bijective. Thus

$$\mathsf{S}^k(\mathbb{C}^n) \cong \mathbb{C}[x_1, \ldots, x_n]_k. \tag{3.5}$$

This justifies the use of the Zariski topology, where the elementary closed subsets are the common zeros of a finite number of homogeneous polynomials [49]. Note that for asymmetric tensors, the same association is not possible (although they can still be associated with polynomials via another bijection). As will be subsequently seen, this identification of symmetric tensors with homogeneous polynomials will allow us to prove some interesting facts about symmetric tensor rank.

We will now proceed to define a useful "inner product" on $\mathbb{C}[x_1, \ldots, x_n]_k$. For any $F, G \in \mathbb{C}[x_1, \ldots, x_n]_k$ written as

$$F(\mathbf{x}) = \sum\nolimits_{|\boldsymbol{p}|=k} \binom{k}{p_1, \ldots, p_n} a_{\boldsymbol{p}} \mathbf{x}^{\boldsymbol{p}}, \qquad G(\mathbf{x}) = \sum\nolimits_{|\boldsymbol{p}|=k} \binom{k}{p_1, \ldots, p_n} b_{\boldsymbol{p}} \mathbf{x}^{\boldsymbol{p}},$$

we let

$$\langle F, G \rangle := \sum\nolimits_{|\boldsymbol{p}|=k} \binom{k}{p_1, \ldots, p_n} a_{\boldsymbol{p}} b_{\boldsymbol{p}} = \sum\nolimits_{p_1+\cdots+p_n=k} \frac{k!}{p_1! \cdots p_n!} a_{p_1 \cdots p_n} b_{p_1 \cdots p_n}.$$

Note that $\langle \cdot, \cdot \rangle$ cannot be an inner product in the usual sense since $\langle F, F \rangle$ is in general complex valued (recall that for an inner product, we will need $\langle F, F \rangle \geq 0$ for all F). However, we will show that it is a nondegenerate symmetric bilinear form.

LEMMA 3.5. *The bilinear form $\langle \cdot, \cdot \rangle : \mathbb{C}[x_1, \ldots, x_n]_k \times \mathbb{C}[x_1, \ldots, x_n]_k \to \mathbb{C}$ defined above is symmetric and nondegenerate. In other words, $\langle F, G \rangle = \langle G, F \rangle$ for every $F, G \in \mathbb{C}[x_1, \ldots, x_n]_k$, and if $\langle F, G \rangle = 0$ for all $G \in \mathbb{C}[x_1, \ldots, x_n]_k$, then $F \equiv 0$.*

Proof. The bilinearity and symmetry is immediate from definition. Suppose $\langle F, G \rangle = 0$ for all $G \in \mathbb{C}[x_1, \ldots, x_n]_k$. Choose G to be the monomials

$$G_{\boldsymbol{p}}(\mathbf{x}) = \binom{k}{p_1, \ldots, p_n} \mathbf{x}^{\boldsymbol{p}},$$

where $|\boldsymbol{p}| = k$, and we see immediately that

$$a_{\boldsymbol{p}} = \langle F, G_{\boldsymbol{p}} \rangle = 0.$$

Thus $F \equiv 0$. ∎

In the special case where G is the kth power of a linear form, we have the following lemma. The main interest in introducing this inner product lies precisely in establishing this lemma.

LEMMA 3.6. *Let $G = (\beta_1 x_1 + \cdots + \beta_n x_n)^k$. Then for any $F \in \mathbb{C}[x_1, \ldots, x_n]_k$, we have*

$$\langle F, G \rangle = F(\beta_1, \ldots, \beta_n),$$

i.e., F evaluated at $(\beta_1, \ldots, \beta_n) \in \mathbb{C}^n$.

Proof. Let $b_{\boldsymbol{p}} = \beta_1^{p_1} \cdots \beta_n^{p_n}$ for all $\boldsymbol{p} = (p_1, \dots, p_n)$ such that $|\boldsymbol{p}| = k$. The multinomial expansion then yields

$$(\beta_1 x_1 + \cdots + \beta_n x_n)^k = \sum\nolimits_{|\boldsymbol{p}|=k} \binom{k}{p_1, \dots, p_n} b_{\boldsymbol{p}} \mathbf{x}^{\boldsymbol{p}}.$$

For any $F(\mathbf{x}) = \sum_{|\boldsymbol{p}|=k} \binom{k}{p_1,\dots,p_n} a_{\boldsymbol{p}} \mathbf{x}^{\boldsymbol{p}}$,

$$F(\beta_1, \dots, \beta_n) = \sum\nolimits_{|\boldsymbol{p}|=k} \binom{k}{p_1, \dots, p_n} a_{\boldsymbol{p}} b_{\boldsymbol{p}} = \langle F, G \rangle$$

as required. ∎

3.2. Equivalence with usual definition. As mentioned earlier, we will show that a tensor is symmetric in the sense of Definition 3.2 if and only if its corresponding array is symmetric in the sense of Definition 3.1.

PROPOSITION 3.7. *Let $A \in \mathsf{T}^k(\mathbb{C}^n)$ and $[\![a_{j_1 \cdots j_k}]\!] \in \mathbb{C}^{n \times \cdots \times n}$ be its corresponding k-array. Then*

$$\sigma(A) = A$$

for all permutations $\sigma \in \mathfrak{S}_k$ if and only if

$$a_{i_{\sigma(1)} \cdots i_{\sigma(k)}} = a_{i_1 \cdots i_k}, \qquad i_1, \dots, i_k \in \{1, \dots, n\},$$

for all permutations $\sigma \in \mathfrak{S}_k$.

Proof. Suppose $[\![a_{i_1 \cdots i_k}]\!] \in \mathbb{C}^{n \times \cdots \times n}$ is symmetric in the sense of Definition 3.1. Then the corresponding tensor

$$A = \sum\nolimits_{i_1, \dots, i_k = 1}^{n} a_{i_1 \cdots i_k} \mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_k},$$

where $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ denotes the canonical basis in $\mathbb{C}^n$, satisfies the following:

$$\begin{aligned}
S(A) &= \sum\nolimits_{i_1, \dots, i_k = 1}^{n} a_{i_1 \cdots i_k} S(\mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_k}) && (S \text{ linear}) \\
&= \frac{1}{k!} \sum\nolimits_{i_1, \dots, i_k = 1}^{n} a_{i_1 \cdots i_k} \Big[\sum\nolimits_{\sigma \in \mathfrak{S}_k} \mathbf{e}_{i_{\sigma(1)}} \otimes \cdots \otimes \mathbf{e}_{i_{\sigma(k)}} \Big] \\
&= \frac{1}{k!} \sum\nolimits_{i_1, \dots, i_k = 1}^{n} \Big[\sum\nolimits_{\sigma \in \mathfrak{S}_k} a_{i_{\sigma(1)} \cdots i_{\sigma(k)}} \Big] \mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_k} \\
&= \frac{1}{k!} \sum\nolimits_{i_1, \dots, i_k = 1}^{n} k! a_{i_1 \cdots i_k} \mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_k} && ([\![a_{i_1 \cdots i_k}]\!] \text{ symmetric}) \\
&= A.
\end{aligned}$$

Hence A is a symmetric tensor in the sense of Definition 3.2.

Conversely, let $A \in \mathsf{T}^k(\mathbb{C}^n)$ be symmetric in the sense of Definition 3.2 and

$$A = \sum\nolimits_{i_1, \dots, i_k = 1}^{n} a_{i_1 \cdots i_k} \mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_k}$$

be the expression of A with respect to $\{\mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_k} \mid 1 \le i_1, \dots i_k \le n\}$, the canonical basis of $\mathsf{T}^k(\mathbb{C}^n)$. Then

$$S(A) = A$$

implies

$$\sum\nolimits_{i_1,\ldots,i_k=1}^{n}\left[\frac{1}{k!}\sum\nolimits_{\sigma\in\mathfrak{S}_k} a_{i_{\sigma(1)}\cdots i_{\sigma(k)}}\right]\mathbf{e}_{i_1}\otimes\cdots\otimes\mathbf{e}_{i_k}=\sum\nolimits_{i_1,\ldots,i_k=1}^{n} a_{i_1\cdots i_k}\mathbf{e}_{i_1}\otimes\cdots\otimes\mathbf{e}_{i_k}.$$

Since $\{\mathbf{e}_{i_1}\otimes\cdots\otimes\mathbf{e}_{i_k} \mid 1\le i_1,\ldots i_k\le n\}$ is a linearly independent set, we must have

$$\frac{1}{k!}\sum\nolimits_{\sigma\in\mathfrak{S}_k} a_{i_{\sigma(1)}\cdots i_{\sigma(k)}}=a_{i_1\cdots i_k}\qquad\text{for all } i_1,\ldots,i_k\in\{1,\ldots,n\}. \tag{3.6}$$

For any given $\tau\in\mathfrak{S}_k$, we have

$$\begin{aligned}
a_{i_{\tau(1)}\cdots i_{\tau(k)}} &= \frac{1}{k!}\sum\nolimits_{\sigma\in\mathfrak{S}_k} a_{i_{\sigma(\tau(1))}\cdots i_{\sigma(\tau(k))}} && \text{(by (3.6))}\\
&= \frac{1}{k!}\sum\nolimits_{\sigma\in\tau\mathfrak{S}_k} a_{i_{\sigma(1)}\cdots i_{\sigma(k)}} &&\\
&= \frac{1}{k!}\sum\nolimits_{\sigma\in\mathfrak{S}_k} a_{i_{\sigma(1)}\cdots i_{\sigma(k)}} && (\tau\mathfrak{S}_k=\mathfrak{S}_k \text{ as } \mathfrak{S}_k \text{ is a group})\\
&= a_{i_1\cdots i_k} && \text{(by (3.6)).}
\end{aligned}$$

Since this holds for arbitrary $\tau\in\mathfrak{S}_k$, the array $[\![a_{i_1\cdots i_k}]\!]$ is symmetric in the sense of Definition 3.1. ∎

4. Notions of rank for symmetric tensors. We will discuss two notions of rank for symmetric tensors—the outer product rank (defined for all tensors) and the symmetric outer product rank (defined only for symmetric tensors). We will show that under certain conditions, they are one and the same. However, it is not known if they are equal on all symmetric tensors in general.

4.1. Outer product decomposition and rank. Any tensor can always be decomposed (possibly nonuniquely) as

$$A=\sum\nolimits_{i=1}^{r}\mathbf{u}_i\otimes\mathbf{v}_i\otimes\cdots\otimes\mathbf{w}_i. \tag{4.1}$$

The *tensor rank*, $\operatorname{rank}(A)$, is defined as the smallest integer r such that this decomposition holds exactly [29, 30]. Among other properties, note that this outer product decomposition remains valid in a ring and that an outer product decomposition of a multilinear transform of A equals the multilinear transform of an outer product decomposition of A. In other words, if (4.1) is an outer product decomposition of A, then

$$A\bullet_1 L\bullet_2 M\bullet_3\cdots\bullet_k N=\sum\nolimits_{i=1}^{r} L\mathbf{u}_i\otimes M\mathbf{v}_i\otimes\cdots\otimes N\mathbf{w}_i$$

is an outer product decomposition of $A\bullet_1 L\bullet_2 M\bullet_3\cdots\bullet_k N$, which may also be written as $(L,M,\ldots,N)\cdot A$. The outer product decomposition has often been regarded synonymously as the data analytic models CANDECOMP [11] and PARAFAC [28], where the decomposition is used to analyze multiway psychometric data.

DEFINITION 4.1. *The rank of* $A=[\![a_{j_1\cdots j_k}]\!]\in\mathbb{C}^{d_1\times\cdots\times d_k}$ *is defined as*

$$\operatorname{rank}(A):=\min\{r \mid A=\textstyle\sum_{i=1}^{r}\mathbf{u}_i\otimes\mathbf{v}_i\otimes\cdots\otimes\mathbf{w}_i\}.$$

If $A=[\![a_{j_1\cdots j_k}]\!]\in\mathsf{S}^k(\mathbb{C}^n)$*, then we may also define the notion of symmetric rank via*

$$\operatorname{rank}_{\mathsf{S}}(A):=\min\{s \mid A=\textstyle\sum_{i=1}^{s}\mathbf{y}_i\otimes\cdots\otimes\mathbf{y}_i\}.$$

Note that over $\mathbb{C}$, the coefficients λ_i appearing in decomposition (1.1) may be set to 1; this is legitimate since any complex number admits a kth root in $\mathbb{C}$. Henceforth, we will adopt the following notation:

$$\mathbf{y}^{\otimes k} := \overbrace{\mathbf{y} \otimes \cdots \otimes \mathbf{y}}^{k \text{ copies}}. \tag{4.2}$$

If in (4.1) we have $\mathbf{u}_i = \mathbf{v}_i = \cdots = \mathbf{w}_i$ for every i, then we may call it a *symmetric outer product decomposition*, yielding a *symmetric rank*, $\operatorname{rank}_{\mathsf{S}}(A)$. Constraints other than full symmetry may be relevant in some application areas, such as partial symmetry as in INDSCAL [11, 58], or positivity/nonnegativity [41, 51, 55].

The definition of symmetric rank is not vacuous because of the following result.

LEMMA 4.2. *Let $A \in \mathsf{S}^k(\mathbb{C}^n)$. Then there exist $\mathbf{y}_1, \ldots, \mathbf{y}_s \in \mathbb{C}^n$ such that*

$$A = \sum\nolimits_{i=1}^{s} \mathbf{y}_i^{\otimes k}.$$

Proof. What we have to prove is that the vector space generated by the kth powers of linear forms $L(\mathbf{x})^k$ (for all $L \in \mathbb{C}^n$) is not included in a hyperplane of $\mathsf{S}^k(\mathbb{C}^n)$. This is indeed true, because otherwise there would exist a nonzero element of $\mathsf{S}^k(\mathbb{C}^n)$ which is orthogonal, under the bilinear form $\langle \cdot, \cdot \rangle$, to all $L(\mathbf{x})^k$ for $L \in \mathbb{C}^n$. Equivalently, by Lemma 3.6, there exists a nonzero polynomial $q(\mathbf{x})$ of degree k such that $q(L) = 0$ for all $L \in \mathbb{C}^n$. But this is impossible, since a nonzero polynomial does not vanish identically on $\mathbb{C}^n$. ∎

Lemma 4.2 may be viewed as a particular case of a basic result in algebraic geometry, stating that the linear space generated by points of an algebraic variety that is not included in a hyperplane, i.e., a subspace of codimension 1, is the whole space [27, 18, 49]. For completeness, a proof of our special case is given above. Note that it follows from the proof that

$$\operatorname{rank}_{\mathsf{S}}(A) \leq \binom{n+k-1}{k}$$

for all $A \in \mathsf{S}^k(\mathbb{C}^n)$.

On the other hand, given a symmetric tensor A, one can compute its outer product decomposition either in $\mathsf{S}^k(\mathbb{C}^n)$ or in $\mathsf{T}^k(\mathbb{C}^n)$. Since the outer product decomposition in $\mathsf{S}^k(\mathbb{C}^n)$ is constrained, it follows that for all $A \in \mathsf{S}^k(\mathbb{C}^n)$,

$$\operatorname{rank}(A) \leq \operatorname{rank}_{\mathsf{S}}(A). \tag{4.3}$$

We will show that equality holds generically when $\operatorname{rank}_{\mathsf{S}}(A) \leq n$ and when k is sufficiently large with respect to n and always holds when $\operatorname{rank}_{\mathsf{S}}(A) = 1, 2$. While we do not know if the equality holds in general, we suspect that this is the case as we are unaware of any counterexample.

4.2. Secant varieties of the Veronese variety. Let us recall here the correspondence between symmetric outer product decompositions and secant varieties of the Veronese variety. By the bijective correspondence between symmetric tensors and homogeneous polynomials established in (3.5), we may discuss this in the context of homogeneous polynomials. The set of homogeneous polynomials that may be written as a kth power of a linear form, $\beta(x)^k = (\beta_1 x_1 + \cdots + \beta_n x_n)^k$ for $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n) \in \mathbb{C}^n$, is a closed algebraic set. We may consider this construction as a map from $\mathbb{C}^n$ to the

space of symmetric tensors given by

$$\begin{aligned}\nu_{n,k} : \mathbb{C}^n &\to \mathbb{C}[x_1, \ldots, x_n]_k \cong \mathsf{S}^k(\mathbb{C}^n),\\ \boldsymbol{\beta} &\mapsto \beta(x)^k.\end{aligned}$$

The image $\nu_{n,k}(\mathbb{C}^n)$ is called the *Veronese variety* and is denoted $\mathcal{V}_{n,k}$ [27, 65]. Following this point of view, a symmetric tensor is of symmetric rank 1 if it corresponds to a point on the Veronese variety. A symmetric tensor is of symmetric rank r if it is a linear combination of r symmetric tensors of symmetric rank 1 but not a linear combination of $r-1$ or fewer such tensors. In other words, a symmetric tensor is of symmetric rank not more than r if it is in the linear space spanned by r points of the Veronese variety. The closure of the union of all linear spaces spanned by r points of the Veronese variety $\mathcal{V}_{n,k}$ is called[1] the $(r-1)$*th-secant variety* of $\mathcal{V}_{n,k}$. See [27, 65] for examples and general properties of these algebraic sets. In the asymmetric case, a corresponding notion is obtained by considering the *Segre variety*, i.e., the image of the Segre map defined in section 2.

4.3. Why rank can exceed dimension. We are now in a position to state and prove the following proposition, which is related to a classical result in algebraic geometry stating that r points in $\mathbb{C}^n$ form the solution set of polynomial equations of degree $\leq r$ [27, p. 6]. This implies that we can find a polynomial of degree $\leq r-1$ that vanishes at $r-1$ of the points L_i but not at the last one, and hence the independence of polynomials $L_1^{r-1}, \ldots, L_r^{r-1}$ follows. Since this proposition is important to our discussion in section 5 (via its corollary), we give a direct and simple proof.

PROPOSITION 4.3. *Let $L_1, \ldots, L_r \in \mathbb{C}[x_1, \ldots, x_n]_1$, i.e., linear forms in n variables. If for all $i \neq j$, L_i is not a scalar multiple of L_j, then for any $k \geq r-1$, the polynomials $L_1^k, \ldots, L_r^k$ are linearly independent in $\mathbb{C}[x_1, \ldots, x_n]$.*

Proof. Let $k \geq r-1$. Suppose that for some $\lambda_1, \ldots, \lambda_r$, $\sum_{i=1}^r \lambda_i L_i^k = 0$. Hence, by the duality property of Lemma 3.6,

$$\sum\nolimits_{i=1}^r \lambda_i \langle F, L_i^k \rangle = \sum\nolimits_{i=1}^r \lambda_i F(L_i) = 0$$

for all $F \in \mathbb{C}[x_1, \ldots, x_n]_k$. Let us prove that we can find a homogeneous polynomial F of degree k that vanishes at $L_1, \ldots, L_{r-1}$ and not at L_r.

Consider a homogeneous polynomial F of degree $k \geq r-1$ that is a multiple of the product of $r-1$ linear forms H_i vanishing at L_i but not at L_r. We have $F(L_r) \neq 0$ but $F(L_j) = 0$, $1 \leq j \leq r-1$. As a consequence, we must have $\lambda_r = 0$. By a similar argument, we may show that $\lambda_i = 0$ for all $i = 1, \ldots, r$. It follows that the polynomials $L_1^k, \ldots, L_r^k$ are linearly independent. ∎

Notice that the bound $r-1$ on the degree can be reduced by d if a d-dimensional linear space containing any $d+1$ of these points does not contain one of the other points [27, p. 6]. In this case, we can replace the product of $d+1$ linear forms H_i vanishing at $d+1$ points by just 1 linear form vanishing at these $d+1$ points.

COROLLARY 4.4. *Let $\mathbf{v}_1, \ldots, \mathbf{v}_r \in \mathbb{C}^n$ be r pairwise linearly independent vectors. For any integer $k \geq r-1$, the rank-1 symmetric tensors*

$$\mathbf{v}_1^{\otimes k}, \ldots, \mathbf{v}_r^{\otimes k} \in \mathsf{S}^k(\mathbb{C}^n)$$

are linearly independent.

[1] This seemingly odd choice, i.e., $r-1$ instead of r, is standard [27, 65] because one wants to be consistent with the usual meaning of a secant, i.e., 1-secant, as a line intersecting *two* points in the variety.

This corollary extends results of [19, Lemma 2.2] and [33, Appendix]. Note that vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$ need not be linearly independent.

Example 4.5. Vectors $\mathbf{v}_1 = (1, 0)$, $\mathbf{v}_2 = (0, 1)$, and $\mathbf{v}_3 = (1, 1)$ are pairwise noncollinear but linearly dependent. According to Corollary 4.4, the symmetric tensors $\mathbf{v}_1^{\otimes k}, \mathbf{v}_2^{\otimes k}, \mathbf{v}_3^{\otimes k}$ are linearly independent for any $k \geq 2$. Evidently, we see that this holds true for $k = 2$ since the matrix below has rank 3:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

4.4. Genericity. Roughly speaking, a property is referred to as *typical* if it holds true on a non-zero-volume set and *generic* if is true almost everywhere. Proper definitions will follow later in section 6. It is important to distinguish between typical and generic properties; for instance, as will be subsequently seen, there can be several *typical ranks* but by definition only a single *generic rank*. We will see that there can be only one typical rank over $\mathbb{C}$, and it is thus generic.

Through the bijection (3.4), the symmetric outer product decomposition (4.1) of symmetric tensors can be carried over to quantics, as pointed out in [16]. The bijection allows one to talk indifferently about the symmetric outer product decomposition of order-k symmetric tensors and the decomposition of degree-k quantics into a sum of linear forms raised to the kth power.

For a long time, it was believed that there was no explicit expression for the generic rank. As Reznick pointed out in [48], Clebsh proved that even when the numbers of free parameters are the same on both sides of the symmetric outer product decomposition, the generic rank may not be equal to $\frac{1}{n}\binom{n+k-1}{k}$. For example, in the case $(k, n) = (4, 3)$, there are $\binom{6}{4} = 15$ degrees of freedom but the generic symmetric rank $R_{\mathsf{S}}(4, 3) = 6 \neq 5 = \frac{1}{3}\binom{6}{4}$. In fact, this holds true over both $\mathbb{R}$ [48] and $\mathbb{C}$ [22]. In section 7, we will see that the generic rank in $\mathsf{S}^k(\mathbb{C}^n)$ is now known for any order and dimension due to the ground breaking work of Alexander and Hirschowitz.

The special case of cubics ($k = 3$) is much better known—a complete classification has been known since 1964, although a constructive algorithm to compute the symmetric outer product decomposition was proposed only recently [35]. The simplest case of binary quantics ($n = 2$) has also been known for more than two decades [62, 16, 38]—a result that is used in real-world engineering problems [15].

5. Rank and symmetric rank. Let $\overline{R}_{\mathsf{S}}(k, n)$ be the generic symmetric rank and $R_{\mathsf{S}}(k, n)$ be the maximally attainable symmetric rank in the space of symmetric tensors $\mathsf{S}^k(\mathbb{C}^n)$. Similarly, let $\overline{R}(k, n)$ be the generic rank and $R(k, n)$ be the maximally attainable rank in the space of order-k dimension-n cubical tensors $\mathsf{T}^k(\mathbb{C}^n)$. Since $\mathsf{S}^k(\mathbb{C}^n)$ is a subspace of $\mathsf{T}^k(\mathbb{C}^n)$, generic and maximal ranks (when they exist) are related for every fixed order k and dimension n as follows:

$$\overline{R}(k, n) \geq \overline{R}_{\mathsf{S}}(k, n) \quad \text{and} \quad R(k, n) \geq R_{\mathsf{S}}(k, n). \tag{5.1}$$

It may seem odd that the inequalities in (5.1) and (4.3) are reversed, but there is no contradiction since the spaces are not the same.

It is then legitimate to ask whether the symmetric rank and the rank are always equal. We show that this holds generically when $\operatorname{rank}_{\mathsf{S}}(A) \leq n$ (Proposition 5.3) or when the order k is sufficiently large relative to the dimension n (Proposition 5.4).

This always holds (not just generically) when $\operatorname{rank}_{\mathsf{S}}(A) = 1, 2$ (Proposition 5.5). We will need some preliminary results in proving these assertions.

LEMMA 5.1. *Let $\mathbf{y}_1, \dots, \mathbf{y}_s \in \mathbb{C}^n$ be linearly independent. Then the symmetric tensor defined by*

$$A := \sum\nolimits_{i=1}^{s} \mathbf{y}_i^{\otimes k}$$

has $\operatorname{rank}_{\mathsf{S}}(A) = s$.

Proof. Suppose $\operatorname{rank}_{\mathsf{S}}(A) = r$. Then there exist $\mathbf{z}_1, \dots, \mathbf{z}_r \in \mathbb{C}^n$ such that

$$\sum\nolimits_{i=1}^{s} \mathbf{y}_i^{\otimes k} = A = \sum\nolimits_{j=1}^{r} \mathbf{z}_j^{\otimes k}. \tag{5.2}$$

By the linear independence of $\mathbf{y}_1, \dots, \mathbf{y}_s$, there exist covectors $\varphi_1, \dots, \varphi_s \in (\mathbb{C}^n)^*$ that are dual to $\mathbf{y}_1, \dots, \mathbf{y}_s$, i.e.,

$$\varphi_i(\mathbf{y}_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Contracting both sides of (5.2) in the first $k-1$ modes with $\varphi_i^{\otimes(k-1)} \in \mathsf{S}^{k-1}((\mathbb{C}^n)^*)$, we get

$$\mathbf{y}_i = \sum\nolimits_{j=1}^{r} \alpha_j \mathbf{z}_j,$$

where $\alpha_j = \varphi_i(\mathbf{z}_j)^{k-1}$. In other words, $\mathbf{y}_i \in \operatorname{span}\{\mathbf{z}_1, \dots, \mathbf{z}_r\}$. Since this holds for each $i = 1, \dots, s$, it implies that the s linearly independent vectors $\mathbf{y}_1, \dots, \mathbf{y}_s$ are contained in $\operatorname{span}\{\mathbf{z}_1, \dots, \mathbf{z}_r\}$. Hence we must have $r \geq s$. On the other hand, it is clear that $r \leq s$. Thus we must have equality. ∎

LEMMA 5.2. *Let $s \leq n$. Let $A \in \mathsf{S}^k(\mathbb{C}^n)$ with* $\operatorname{rank}_{\mathsf{S}}(A) = s$ *and*

$$A = \sum\nolimits_{i=1}^{s} \mathbf{y}_i^{\otimes k}$$

be a symmetric outer product decomposition of A. Then vectors of the set $\{\mathbf{y}_1, \dots, \mathbf{y}_s\}$ are generically linearly independent.

Proof. We will write

$$\mathcal{Y}_s := \{A \in \mathsf{S}^k(\mathbb{C}^n) \mid \operatorname{rank}_{\mathsf{S}}(A) \leq s\} \quad \text{and} \quad \mathcal{Z}_s := \{A \in \mathsf{S}^k(\mathbb{C}^n) \mid \operatorname{rank}_{\mathsf{S}}(A) = s\}.$$

Define the map from the space of $n \times s$ matrices to order-k symmetric tensors,

$$f : \mathbb{C}^{n \times s} \to \mathsf{S}^k(\mathbb{C}^n),$$
$$[\mathbf{y}_1, \dots, \mathbf{y}_s] \mapsto \sum\nolimits_{i=1}^{s} \mathbf{y}_i^{\otimes k}.$$

It is clear that f takes $\mathbb{C}^{n \times s}$ *onto* $\mathcal{Y}_s$ (i.e., $f(\mathbb{C}^{n \times s}) = \mathcal{Y}_s$). We let E_0 and E_1 be the subsets of rank-deficient and full-rank matrices in $\mathbb{C}^{n \times s}$, respectively. Thus we have the disjoint union

$$E_0 \cup E_1 = \mathbb{C}^{n \times s}, \quad E_0 \cap E_1 = \varnothing.$$

Recall that the full-rank matrices are generic in $\mathbb{C}^{n \times s}$. Recall also that E_0 is an algebraic set in $\mathbb{C}^{n \times s}$ defined by the vanishing of all $s \times s$ principal minors. By the previous lemma, $f(E_1) \subseteq \mathcal{Z}_s$. The set of symmetric tensors

$$\sum\nolimits_{i=1}^{s} \mathbf{y}_i^{\otimes k}$$

in $\mathcal{Z}_s$ for which $\{\mathbf{y}_1, \dots, \mathbf{y}_s\}$ is linearly dependent, i.e., $[\mathbf{y}_1, \dots, \mathbf{y}_s]$ is rank deficient, is simply

$$\mathcal{Z}_s \cap f(E_0).$$

Since f is a polynomial map and E_0 is a nontrivial algebraic set, we conclude that $f(E_1)$ is generic in $\mathcal{Z}_s$. □

PROPOSITION 5.3. *Let* $A \in \mathsf{S}^k(\mathbb{C}^n)$. *If* $\operatorname{rank}_{\mathsf{S}}(A) \leq n$, *then* $\operatorname{rank}(A) = \operatorname{rank}_{\mathsf{S}}(A)$ *generically.*

Proof. Let $r = \operatorname{rank}(A)$ and $s = \operatorname{rank}_{\mathsf{S}}(A)$. There exist decompositions

$$\sum\nolimits_{j=1}^{r} \mathbf{x}_j^{(1)} \otimes \cdots \otimes \mathbf{x}_j^{(k)} = A = \sum\nolimits_{i=1}^{s} \mathbf{y}_i^{\otimes k}. \tag{5.3}$$

By Lemma 5.2, we may assume that for a generic $A \in \mathcal{Z}_s$, the vectors $\mathbf{y}_1, \dots, \mathbf{y}_s$ are linearly independent. As in the proof of Lemma 5.1, we may find a set of covectors $\varphi_1, \dots, \varphi_s \in (\mathbb{C}^n)^*$ that are dual to $\mathbf{y}_1, \dots, \mathbf{y}_s$, i.e.,

$$\varphi_i(\mathbf{y}_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Contracting both sides of (5.3) in the first $k-1$ modes with $\varphi_i^{\otimes(k-1)} \in \mathsf{S}^{k-1}((\mathbb{C}^n)^*)$, we get

$$\sum\nolimits_{j=1}^{r} \alpha_{ij} \mathbf{x}_j^{(k)} = \mathbf{y}_i,$$

where $\alpha_{ij} = \varphi_i(\mathbf{x}_j^{(1)}) \cdots \varphi_i(\mathbf{x}_j^{(k-1)})$, $j = 1, \dots, r$. Since this holds for each $i = 1, \dots, s$, it implies that the s linearly independent vectors $\mathbf{y}_1, \dots, \mathbf{y}_s$ are contained in $\operatorname{span}\{\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_r^{(k)}\}$. Hence we must have $r \geq s$. On the other hand, it is clear that $r \leq s$. Thus we must have equality. □

We will see below that we could have $\operatorname{rank}(A) = \operatorname{rank}_{\mathsf{S}}(A)$ even when the constituting vectors $\mathbf{y}_1, \dots, \mathbf{y}_s$ are not linearly independent. The authors would like to thank David Gross for his help in correcting an error in the original proof.

PROPOSITION 5.4. *Let* $\mathbf{y}_1, \dots, \mathbf{y}_s \in \mathbb{C}^n$ *be pariwise linearly independent. If* k *is sufficiently large, then the symmetric tensor defined by*

$$A := \sum\nolimits_{i=1}^{s} \mathbf{y}_i^{\otimes k}$$

satisfies $\operatorname{rank}(A) = \operatorname{rank}_{\mathsf{S}}(A)$ *generically.*

Proof. Let $r = \operatorname{rank}(A)$ and $s = \operatorname{rank}_{\mathsf{S}}(A)$. So there exist decompositions

$$\sum\nolimits_{j=1}^{r} \mathbf{x}_j^{(1)} \otimes \cdots \otimes \mathbf{x}_j^{(k)} = A = \sum\nolimits_{i=1}^{s} \mathbf{y}_i^{\otimes k}. \tag{5.4}$$

Note that the left-hand side may be written $\sum_{i=1}^{s} \mathbf{y}_i^{\otimes k/2} \otimes \mathbf{y}_i^{\otimes k/2}$, where we have assumed, without loss of generality, that k is even. By Proposition 4.3, when k is sufficiently large, the order-$(k/2)$ tensors $\mathbf{y}_1^{\otimes k/2}, \dots, \mathbf{y}_s^{\otimes k/2}$ are generically linearly independent. Hence we may find functionals $\Phi_1, \dots, \Phi_s \in \mathsf{S}^{k/2}(\mathbb{C}^n)^*$ that are dual to $\mathbf{y}_1^{\otimes k/2}, \dots, \mathbf{y}_s^{\otimes k/2} \in \mathsf{S}^{k/2}(\mathbb{C}^n)$, i.e.,

$$\Phi_i(\mathbf{y}_j^{\otimes k/2}) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Contracting both sides of (5.4) in the first $k/2$ modes with Φ_i, we get

$$\sum\nolimits_{j=1}^{r} \alpha_{ij} \mathbf{x}_j^{(k/2+1)} \otimes \cdots \otimes \mathbf{x}_j^{(k)} = \mathbf{y}_i^{\otimes k/2},$$

where $\alpha_{ij} = \Phi_i(\mathbf{x}_j^{(1)} \otimes \cdots \otimes \mathbf{x}_j^{(k/2)})$, $j = 1, \ldots, r$. Since this holds for each $i = 1, \ldots, s$, it implies that the s linearly independent vectors $\mathbf{y}_1, \ldots, \mathbf{y}_s$ are contained in $\operatorname{span}\{\mathbf{x}_1^{(k/2+1)} \otimes \cdots \otimes \mathbf{x}_1^{(k)}, \ldots \mathbf{x}_r^{(k/2+1)} \otimes \cdots \otimes \mathbf{x}_r^{(k)}\}$. Hence we must have $r \geq s$. On the other hand, it is clear that $r \leq s$. Thus we must have equality. ∎

PROPOSITION 5.5. *Let* $A \in \mathsf{S}^k(\mathbb{C}^n)$. *If* $\operatorname{rank}_{\mathsf{S}}(A) = 1$ *or* 2, *then* $\operatorname{rank}(A) = \operatorname{rank}_{\mathsf{S}}(A)$.

Proof. If $\operatorname{rank}_{\mathsf{S}}(A) = 1$, then $\operatorname{rank}(A) = 1$ clearly. If $\operatorname{rank}_{\mathsf{S}}(A) = 2$, then

$$A = \mathbf{y}_1^{\otimes k} + \mathbf{y}_2^{\otimes k}$$

for some $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{C}^n$. It is clear that $\mathbf{y}_1$ and $\mathbf{y}_2$ must be linearly independent or otherwise $\mathbf{y}_2 = \alpha \mathbf{y}_1$ implies that

$$A = (\beta \mathbf{y}_1)^{\otimes k}$$

for any $\beta = (1 + \alpha^k)^{1/k}$, contradicting $\operatorname{rank}_{\mathsf{S}}(A) = 2$. It follows from the argument in the proof of Proposition 5.3 with $s = 2$ that $\operatorname{rank}(A) = 2$. ∎

The following result will be useful later.

PROPOSITION 5.6. *Let* $\mathbf{v}_1$ *and* $\mathbf{v}_2$ *be two linearly independent vectors in* $\mathbb{C}^n$. *Then for any* $k > 1$, *the order-*k *symmetric tensor*

$$\begin{aligned}(5.5)\quad &\mathbf{v}_1 \otimes \mathbf{v}_2 \otimes \mathbf{v}_2 \otimes \cdots \otimes \mathbf{v}_2 + \mathbf{v}_2 \otimes \mathbf{v}_1 \otimes \mathbf{v}_2 \otimes \cdots \otimes \mathbf{v}_2 \\ &\quad + \mathbf{v}_2 \otimes \mathbf{v}_2 \otimes \mathbf{v}_1 \otimes \cdots \otimes \mathbf{v}_2 + \cdots + \mathbf{v}_2 \otimes \mathbf{v}_2 \otimes \mathbf{v}_2 \otimes \cdots \otimes \mathbf{v}_1\end{aligned}$$

is of symmetric rank k.

Proof. It is not hard to check that the symmetric tensor in (5.5) is associated with the quantic $p(z_1, z_2) = z_1 z_2^{k-1}$ up to a constant multiplicative factor (where z_1, z_2 are the first two coordinate variables in $(z_1, \ldots, z_n)$).

To prove that this quantic is of symmetric rank k, we are going to show that $p(z_1, z_2)$ can be decomposed into a sum of powers of linear forms as

$$(5.6)\qquad p(z_1, z_2) = \sum\nolimits_{i=1}^{k} \lambda_i (\alpha_i z_1 + \beta_i z_2)^k.$$

There are infinitely many possibilities of choosing coefficients (α_i, β_i) but we need to provide just one solution. Take $\alpha_1 = \cdots = \alpha_r = 1$ and $\beta_1, \ldots, \beta_k$ distinct such that

$$(5.7)\qquad \sum\nolimits_{i=1}^{k} \beta_i = 0.$$

First we express all quantics in terms of the canonical basis scaled by the binomial coefficients:

$$\{z_1^k, k z_1^{k-1} z_2, \ldots, k z_1 z_2^{k-1}, z_2^k\}.$$

In this basis, the monomial $k z_1 z_2^{k-1}$ can be represented by a $(k+1)$-dimensional vector containing only one nonzero entry. The quantic $(z_i + \beta_i z_2)^k$ is then represented by the vector

$$[1, \beta_i, \beta_i^2, \ldots, \beta_i^k] \in \mathbb{C}^{k+1}.$$

The existence of coefficients $\lambda_1, \dots, \lambda_k$ such that we have the decomposition (5.6) is equivalent to the vanishing of the $(k+1) \times (k+1)$ determinant

$$\begin{vmatrix} 0 & 0 & \cdots & 1 & 0 \\ 1 & \beta_1 & \cdots & \beta_1^{k-1} & \beta_1^k \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & \beta_k & \cdots & \beta_k^{k-1} & \beta_k^k \end{vmatrix}. \tag{5.8}$$

An explicit computation shows that this determinant is $\pm(\sum_{i=1}^k \beta_i) V_k(\beta_1, \dots, \beta_k)$, where $V_k(\beta_1, \dots, \beta_k)$ is the Vandermonde determinant of degree $k-1$ of $\beta_1, \dots, \beta_k$. Thus by (5.7), the determinant in (5.8) vanishes.

This proves that the symmetric rank of $z_1 z_2^k$ is $\leq k$. Note that the symmetric rank cannot be smaller than k because removing any row of the matrix of (5.8) still yields a matrix of rank k if the β_i are distinct (see also Proposition 4.3). □

This proof is constructive and gives an algorithm to compute a symmetric outer product decomposition of any binary symmetric tensor of the form (5.5). For example, the reader can check out that the decompositions below may be obtained this way.

Example 5.7. The quantics $48 z_1^3 z_2$ and $60 z_1^4 z_2$ are associated with the symmetric tensors of maximal rank A_{31} and A_{41}, respectively. Their symmetric outer product decompositions are given by

$$\begin{aligned} A_{31} &= 8(\mathbf{v}_1 + \mathbf{v}_2)^{\otimes 4} - 8(\mathbf{v}_1 - \mathbf{v}_2)^{\otimes 4} - (\mathbf{v}_1 + 2\mathbf{v}_2)^{\otimes 4} + (\mathbf{v}_1 - 2\mathbf{v}_2)^{\otimes 4}, \\ A_{41} &= 8(\mathbf{v}_1 + \mathbf{v}_2)^{\otimes 5} - 8(\mathbf{v}_1 - \mathbf{v}_2)^{\otimes 5} - (\mathbf{v}_1 + 2\mathbf{v}_2)^{\otimes 5} + (\mathbf{v}_1 - 2\mathbf{v}_2)^{\otimes 5} + 48\mathbf{v}_1^{\otimes 5}. \end{aligned}$$

The maximal symmetric rank achievable by symmetric tensors of order k and dimension $n = 2$ is k, i.e., $R_{\mathsf{S}}(k, 2) = k$. One can say that such symmetric tensors lie on a tangent line to the Veronese variety of symmetric rank-1 tensors. In [13], an algorithm was proposed to decompose binary forms when their rank is not larger than $k/2$; however, this algorithm would not have found the decompositions above since the symmetric ranks of A_{31} and A_{41} exceed $4/2$ and $5/2$, respectively.

6. Generic symmetric rank and typical symmetric ranks. For given order and dimension, define the following subsets of symmetric tensors $\mathcal{Y}_r := \{A \in \mathsf{S}^k(\mathbb{C}^n) \mid \operatorname{rank}_{\mathsf{S}}(A) \leq r\}$ and $\mathcal{Z}_r := \{A \in \mathsf{S}^k(\mathbb{C}^n) \mid \operatorname{rank}_{\mathsf{S}}(A) = r\}$. Also, denote the corresponding Zariski closures by $\overline{\mathcal{Y}}_r$ and $\overline{\mathcal{Z}}_r$, respectively. Recall that the Zariski closure [18] of a set $\mathcal{S}$ is simply the smallest variety containing $\mathcal{S}$. For every $r \in \mathbb{N}$, we clearly have

$$\mathcal{Y}_{r-1} \cup \mathcal{Z}_r = \mathcal{Y}_r \quad \text{and} \quad \underbrace{\mathcal{Y}_1 + \cdots + \mathcal{Y}_1}_{r \text{ copies}} = \mathcal{Y}_r.$$

The quantities $\overline{R}_{\mathsf{S}}(k, n)$ and $R_{\mathsf{S}}(k, n)$ may now be formally defined by

$$\overline{R}_{\mathsf{S}}(k, n) := \min\{r \mid \overline{\mathcal{Y}}_r = \mathsf{S}^k(\mathbb{C}^n)\} \quad \text{and} \quad R_{\mathsf{S}}(k, n) := \min\{r \mid \mathcal{Y}_r = \mathsf{S}^k(\mathbb{C}^n)\}.$$

By definition, we have $\overline{R}_{\mathsf{S}}(k, n) \leq R_{\mathsf{S}}(k, n)$. We shall prove in this section that a generic symmetric rank always exists in $\mathsf{S}^k(\mathbb{C}^n)$, i.e., there is an r such that $\overline{\mathcal{Z}}_r = \mathsf{S}^k(\mathbb{C}^n)$, and that it is equal to $\overline{R}_{\mathsf{S}}(k, n)$, thus justifying our naming $\overline{R}_{\mathsf{S}}(k, n)$ the *generic* symmetric rank in section 5.

An integer r is not a *typical rank* if $\mathcal{Z}_r$ has zero volume, which means that $\mathcal{Z}_r$ is contained in a nontrivial closed set. This definition is somewhat unsatisfactory since

any mention of "volume" necessarily involves a choice of measure, which is really irrelevant here. A better definition is as follows.

DEFINITION 6.1. *An integer r is a typical rank if $\mathcal{Z}_r$ is dense with the Zariski topology, i.e., if $\overline{\mathcal{Z}}_r = \mathsf{S}^k(\mathbb{C}^n)$. When a typical rank is unique, it may be called generic.*

We used the wording "typical" in agreement with previous terminologies [9, 56, 58]. Since two dense algebraic sets always intersect over $\mathbb{C}$, there can be only one typical rank over $\mathbb{C}$, and hence it is generic. In the remainder of this section, we will write $\overline{R}_{\mathsf{S}} = \overline{R}_{\mathsf{S}}(k,n)$ and $R_{\mathsf{S}} = R_{\mathsf{S}}(k,n)$. We can then prove the following.

PROPOSITION 6.2. *The varieties $\overline{\mathcal{Z}}_r$ can be ordered by inclusion as follows. If*

$$r_1 < r_2 < \overline{R}_{\mathsf{S}} < r_3 \leq R_{\mathsf{S}},$$

then

$$\overline{\mathcal{Z}}_{r_1} \subsetneq \overline{\mathcal{Z}}_{r_2} \subsetneq \overline{\mathcal{Z}}_{\overline{R}_{\mathsf{S}}} \supsetneq \overline{\mathcal{Z}}_{r_3}.$$

Before proving this proposition, we state two preliminary results. Recall that an algebraic variety is *irreducible* if it cannot be decomposed as the union of proper subvarieties (cf. [27, pp. 51] and [49, pp. 34]). In algebraic geometry, it is known that the secant varieties of any irreducible variety are irreducible. Nevertheless, we will give a short proof of the following lemma for the sake of completeness.

LEMMA 6.3. *The sets $\overline{\mathcal{Y}}_r$, $r \geq 1$, are irreducible algebraic varieties.*

Proof. For $r \geq 1$, the variety $\overline{\mathcal{Y}}_r$ is the closure of the image $\mathcal{Y}_r$ of the map

$$\begin{aligned} \varphi_r : \mathbb{C}^{n\times r} &\to \mathsf{S}^k(\mathbb{C}^n), \\ [\mathbf{u}_1, \ldots, \mathbf{u}_r] &\mapsto \sum\nolimits_{i=1}^{r} \mathbf{u}_i^{\otimes k}. \end{aligned}$$

Consider now two polynomials f, g such that $fg \equiv 0$ on $\overline{\mathcal{Y}}_r$. As $\overline{\mathcal{Y}}_r$ is the Zariski closure of $\mathcal{Y}_r$, this is equivalent to $fg \equiv 0$ on $\mathcal{Y}_r$ or

$$(fg) \circ \varphi_r = (f \circ \varphi_r)(g \circ \varphi_r) \equiv 0.$$

Thus either $f \equiv 0$ or $g \equiv 0$ on $\mathcal{Y}_r$ or equivalently on $\overline{\mathcal{Y}}_r$, which proves that $\overline{\mathcal{Y}}_r$ is an irreducible variety. For more details on properties of parameterized varieties, see [18]. See also the proof of [52, 9] for third order tensors. ∎

LEMMA 6.4. *We have $\overline{R}_{\mathsf{S}} = \min\{r \mid \overline{\mathcal{Y}}_r = \overline{\mathcal{Y}}_{r+1}\}$.*

Proof. Suppose that there exists $r < \overline{R}_{\mathsf{S}}$ such that $\overline{\mathcal{Y}}_r = \overline{\mathcal{Y}}_{r+1}$. Then since $\overline{\mathcal{Y}}_r \subseteq \overline{\mathcal{Y}}_r + \mathcal{Y}_1 \subseteq \overline{\mathcal{Y}}_{r+1} = \overline{\mathcal{Y}}_r$, we have

$$\overline{\mathcal{Y}}_r = \overline{\mathcal{Y}}_r + \mathcal{Y}_1 = \overline{\mathcal{Y}}_r + \mathcal{Y}_1 + \mathcal{Y}_1 = \cdots = \overline{\mathcal{Y}}_r + \mathcal{Y}_1 + \cdots + \mathcal{Y}_1.$$

As the sum of R_{S} copies of $\mathcal{Y}_1$ is $\mathsf{S}^k(\mathbb{C}^n)$, we deduce that $\overline{\mathcal{Y}}_r = \mathsf{S}^k(\mathbb{C}^n)$ and thus $r \geq \overline{R}_{\mathsf{S}}$, which contradicts our hypothesis. By definition, $\overline{\mathcal{Y}}_{\overline{R}_{\mathsf{S}}} = \overline{\mathcal{Y}}_{\overline{R}_{\mathsf{S}}+1} = \mathsf{S}^k(\mathbb{C}^n)$, which proves the lemma. See also the proof of [52] for the asymmetric case. ∎

We are now in a position to prove Proposition 6.2.

Proof of Proposition 6.2. By Lemma 6.4, we deduce that for $r < \overline{R}_{\mathsf{S}}$,

$$\overline{\mathcal{Y}}_r \neq \overline{\mathcal{Y}}_{r+1}.$$

As $\overline{\mathcal{Y}}_r$ is an irreducible variety, we have $\dim(\overline{\mathcal{Y}}_r) < \dim(\overline{\mathcal{Y}}_{r+1})$. As $\mathcal{Y}_r \cup \mathcal{Z}_{r+1} = \mathcal{Y}_{r+1}$, we deduce that

$$\overline{\mathcal{Y}}_r \cup \overline{\mathcal{Z}}_{r+1} = \overline{\mathcal{Y}}_{r+1},$$

which implies by the irreducibility of $\overline{\mathcal{Y}}_{r+1}$, that $\overline{\mathcal{Z}}_{r+1} = \overline{\mathcal{Y}}_{r+1}$. Consequently, for $r_1 < r_2 < \overline{R}_{\mathsf{S}}$, we have

$$\overline{\mathcal{Z}}_{r_1} = \overline{\mathcal{Y}}_{r_1} \subsetneq \overline{\mathcal{Z}}_{r_2} = \overline{\mathcal{Y}}_{r_2} \subsetneq \overline{\mathcal{Z}}_{\overline{R}_{\mathsf{S}}} = \overline{\mathcal{Y}}_{\overline{R}_{\mathsf{S}}} = \mathsf{S}^k(\mathbb{C}^n).$$

Let us prove now that if $\overline{R}_{\mathsf{S}} < r_3$, we have $\overline{\mathcal{Z}}_{r_3} \subsetneq \mathsf{S}^k(\mathbb{C}^n)$. Suppose that $\overline{\mathcal{Z}}_{r_3} = \mathsf{S}^k(\mathbb{C}^n)$, then $\mathcal{Z}_{r_3}$ is dense in $\mathsf{S}^k(\mathbb{C}^n)$ as well as $\mathcal{Z}_{\overline{R}_{\mathsf{S}}}$ in the Zariski topology. This implies that $\mathcal{Z}_{r_3} \cap \mathcal{Z}_{\overline{R}_{\mathsf{S}}} \neq \varnothing$, which is false because a tensor cannot have two different ranks. Consequently, we have $\overline{\mathcal{Z}}_{r_3} \subsetneq \mathsf{S}^k(\mathbb{C}^n)$. ∎

PROPOSITION 6.5. *If $1 \leq r \leq R_{\mathsf{S}}$, then $\mathcal{Z}_r \neq \overline{\mathcal{Z}}_r$.*

Proof. Let $r > 1$ and $A \in \mathcal{Z}_r$. Then by definition of $\mathcal{Y}_r$, there exists $A_0 \in \mathcal{Y}_{r-1}$ and $A_1 \in \mathcal{Y}_1$ such that $A = A_0 + A_1$. As $A_0 \notin \mathcal{Y}_{r-2}$ (otherwise $A \in \mathcal{Y}_{r-1}$) we have $A_0 \in \mathcal{Z}_{r-1}$. For $\varepsilon \neq 0$, define $A_\varepsilon = A_0 + \varepsilon A_1$. We have that $A_\varepsilon \in \mathcal{Z}_r$ for all $\varepsilon \neq 0$ and $\lim_{\varepsilon \to 0} A_\varepsilon = A_0$. This shows that $A_0 \in \overline{\mathcal{Z}}_r - \mathcal{Z}_r$ and consequently that $\mathcal{Z}_r \neq \overline{\mathcal{Z}}_r$. ∎

The above proposition is about the set of symmetric tensors of symmetric rank *exactly* r. But what about those of symmetric rank *at most* r? While $\mathcal{Y}_1$ is closed as a determinantal variety, we will see from Examples 6.6 and 6.7 as well as Proposition 6.8 that $\mathcal{Y}_r$ is generally not closed for $r > 1$. This is another major difference from matrices, for which all $\mathcal{Y}_r$ are closed sets.

Example 6.6. In dimension $n \geq 2$, and for any order $k > 2$, $\mathcal{Y}_2$ is not closed. In fact, take two independent vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ and define the sequence of symmetric tensors

$$A_\varepsilon(i,j) := \frac{1}{\varepsilon}\left[(\mathbf{x}_i + \varepsilon \mathbf{x}_j)^{\otimes k} - \mathbf{x}_i^{\otimes k}\right]. \tag{6.1}$$

For any $\varepsilon \neq 0$, $A_\varepsilon(i,j)$ is of symmetric rank 2, but converges in the limit as $\varepsilon \to 0$ to a symmetric tensor of symmetric rank k. In fact, the limiting symmetric tensor is easily seen to be a sum of k rank-1 tensors,

$$\mathbf{x}_i \otimes \mathbf{x}_j \otimes \cdots \otimes \mathbf{x}_j + \mathbf{x}_j \otimes \mathbf{x}_i \otimes \cdots \otimes \mathbf{x}_j + \cdots + \mathbf{x}_j \otimes \mathbf{x}_j \otimes \cdots \otimes \mathbf{x}_i,$$

which has symmetric rank k by Proposition 5.6.

Example 6.7. Let $n = 3$ and $k = 3$. Then $\mathcal{Y}_5 \subset \overline{\mathcal{Y}}_3$, whereas $3 < \overline{R}_{\mathsf{S}}$. In fact, take the symmetric tensor associated with the ternary cubic $p(x,y,z) = x^2y - xz^2$. According to [16, 47], this tensor has rank 5. On the other hand, it is the limit of the sequence $p_\varepsilon(x,y,z) = x^2y - xz^2 + \varepsilon z^3$ as ε tends to zero. According to a result in [16], the latter polynomial is associated with a rank-3 tensor since the determinant of its Hessian is equal to $8x^2(x - 3\varepsilon z)$ and hence contains two distinct linear forms as long as $\varepsilon \neq 0$.

It is easy to show that this lack of closeness extends in general to $r > \overline{R}_{\mathsf{S}}$ or for $r \leq n$, as stated in the two propositions below.

PROPOSITION 6.8. *If $\overline{R}_{\mathsf{S}} < r$, then for all $k > 2$, $\mathcal{Y}_r \neq \overline{\mathcal{Y}}_r$.*

Proof. If $\overline{R}_{\mathsf{S}} < r$, then $\mathcal{Y}_{\overline{R}_{\mathsf{S}}} \subsetneq \mathcal{Y}_r$. By the definition of generic symmetric rank, $\overline{\mathcal{Y}}_{\overline{R}_{\mathsf{S}}} = \mathsf{S}^k(\mathbb{C}^n) = \overline{\mathcal{Y}}_r$. Hence $\mathcal{Y}_r \subsetneq \overline{\mathcal{Y}}_r = \mathsf{S}^k(\mathbb{C}^n)$. ∎

PROPOSITION 6.9. *If $1 < r \leq n$, then for any $k > 2$, $\mathcal{Y}_r \neq \overline{\mathcal{Y}}_r$.*

Proof. Take n linearly independent vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Then the symmetric tensors $\mathbf{x}_1^{\otimes k}, \ldots, \mathbf{x}_n^{\otimes k}$ are linearly independent as well, and $\sum_{i=1}^r \mathbf{x}_i^{\otimes k}$ is of symmetric rank r for every $r \leq n$ by Lemma 5.1. Now for $r > 2$ and any $\varepsilon \neq 0$, define the symmetric tensor

$$A_\varepsilon = \frac{1}{\varepsilon}\left[(\mathbf{x}_1 + \varepsilon \mathbf{x}_2)^{\otimes k} - \mathbf{x}_1^{\otimes k}\right] + \sum\nolimits_{i=3}^{r} \mathbf{x}^{\otimes k}.$$

A_ε is again of symmetric rank r for every $\varepsilon \neq 0$, but tends to a symmetric rank $r+1$ tensor (see also section 8.1). For $r = 2$, the same reasoning applies with

$$A_\varepsilon = \frac{1}{\varepsilon}\left[(\mathbf{x}_1 + \varepsilon \mathbf{x}_2)^{\otimes k} - \mathbf{x}_1^{\otimes k}\right].$$

This shows that $\mathcal{Y}_r$ is not closed. ∎

Based on these two propositions, we conjecture the stronger statement that for order $k > 2$, the set of symmetric tensors of symmetric rank at most r is never closed, even for $r = n+1, \ldots, R_{\mathsf{S}} - 1$.

CONJECTURE 6.10. *Assume $k > 2$ and $n \geq 2$. Then $\mathcal{Y}_r \neq \overline{\mathcal{Y}}_r$ for any r such that $1 < r < R_{\mathsf{S}}$.*

Up to this point, our study has been based on the Zariski topology [49, 18]. However it is useful from a practical point of view to be able to apply these results to other topologies, for example, the Euclidean topology. Since the $\mathcal{Y}_r$ are parameterized and are thus algebraic constructible sets [49], and since the closure of an algebraic constructible set for the Euclidean topology and the Zariski topology are the same, the results in this paper holds true for many other topologies. We have in particular the following result.

COROLLARY 6.11. *Let μ be a measure on Borel subsets of $\mathsf{S}^k(\mathbb{C}^n)$ with respect to the Euclidean topology on $\mathsf{S}^k(\mathbb{C}^n)$. Let $\overline{R}_{\mathsf{S}}$ be the generic symmetric rank in $\mathsf{S}^k(\mathbb{C}^n)$. If μ is absolutely continuous with respect to the Lebesgue measure on $\mathsf{S}^k(\mathbb{C}^n)$, then*

$$\mu(\{A \in \mathsf{S}^k(\mathbb{C}^n) \mid \operatorname{rank}_{\mathsf{S}}(A) \neq \overline{R}_{\mathsf{S}}\}) = 0.$$

In particular, this corollary tells us that $\mathcal{Z}_{\overline{R}_{\mathsf{S}}}$ is also dense in $\mathsf{S}^k(\mathbb{C}^n)$ with respect to the Euclidean topology. It also tells us that the rank of a tensor whose entries are drawn randomly according to an absolutely continuous distribution (e.g., Gaussian) is $\overline{R}_{\mathsf{S}}$ with probability 1. This is useful in signal processing, for instance, where cumulant tensors are estimated from actual data and are asymptotically Gaussian distributed [6, 44].

These statements extend previous results [3] and prove that there can be only *one* subset $\mathcal{Z}_r$ of nonempty interior and that the latter is dense in $\mathsf{S}^k(\mathbb{C}^n)$; this result, however, requires that we work over an algebraically closed field such as $\mathbb{C}$.

The results of this section are indeed not generally valid over $\mathbb{R}$. We refer the reader to section 8 for further discussions concerning the real field.

7. Values of the generic symmetric rank. In practice, it would be useful to be able to compute the symmetric rank of any given symmetric tensor, or at least to know the maximal values of the symmetric rank, given its order and dimensions. Unfortunately, these problems are far from resolved.

The corresponding problem for the generic values of the symmetric rank, however, has seen enormous progress due to the work of Alexander and Hirschowitz described in section 7.1. In fact, even before their breakthrough, bounds on the generic symmetric rank have been known for decades [3, 47, 48]:

$$\left\lceil \frac{1}{n}\binom{n+k-1}{k} \right\rceil \leq \overline{R}_{\mathsf{S}}(k,n) \leq \binom{n+k-2}{k-1}.$$

It is known that the lower bound is often accurate but the upper bound is not tight [16]. Furthermore, exact results are known in the case of binary quantics ($n = 2$) and ternary cubics ($k = 3$) [22, 16, 48, 35].

7.1. Alexander–Hirschowitz theorem. It was not until the work [1] of Alexander and Hirschowitz in 1995 that the generic symmetric rank problem was completely settled. Nevertheless, the relevance of their result has remained largely unknown in the applied and computational mathematics communities. One reason is that the connection between our problem and the *interpolating polynomials* discussed in [1] is not at all well known in the aforementioned circles. So for the convenience of our readers, we will state the result of Alexander and Hirschowitz in the context of the symmetric outer product decomposition below.

THEOREM 7.1 (Alexander–Hirschowitz). *For $k > 2$, the generic symmetric rank of an order-k symmetric tensor of dimension n over $\mathbb{C}$ is always equal to the lower bound*

$$\overline{R}_{\mathsf{S}}(k,n) = \left\lceil \frac{1}{n}\binom{n+k-1}{k} \right\rceil \tag{7.1}$$

except for the following cases: $(k,n) \in \{(3,5),(4,3),(4,4),(4,5)\}$, where it should be increased by 1.

This theorem is extremely complicated to prove, and the interested reader should refer to the two papers by Alexander and Hirschowitz [1, 2]. Simplifications to this proof have also been recently proposed in [12]. It is worth noting that these results have been proved in terms of multivariate polynomials and interpolation theory, not in terms of symmetric tensors. The exception $(k,n) = (4,3)$ has been known since 1860; in fact, Sylvester referred to it as the Clebsch theorem in his work [53]. It is not hard to guess the formula in (7.1) by a degrees-of-freedom argument. The difficulty of proving Theorem 7.1 lies in establishing the fact that the four given exceptions to the expected formula (7.1) are the only ones. Table 7.1 lists a few values of the generic symmetric rank.

TABLE 7.1
Values of the generic symmetric rank $\overline{R}_{\mathsf{S}}(k,n)$ for various orders k and dimensions n. Values appearing in bold are the exceptions outlined by the Alexander–Hirschowitz theorem.

k \ n	2	3	4	5	6	7	8	9	10
3	2	4	5	**8**	10	12	15	19	22
4	3	**6**	**10**	**15**	21	30	42	55	72
5	3	7	14	26	42	66	99	143	201
6	4	10	21	42	77	132	215	334	501

7.2. Uniqueness. Besides the exceptions pointed out in Theorem 7.1, the number of solutions for the symmetric outer product decomposition has to be finite if the rank r is smaller than or equal to $\frac{1}{n}\binom{n+k-1}{k}$. This occurs, for instance, for all cases of degree $k = 5$ in Table 7.1, except for $n = 5$ and $n = 10$. Hence we may deduce the following.

COROLLARY 7.2. *Suppose $(k,n) \notin \{(3,5),(4,3),(4,4),(4,5)\}$. Let $A \in \mathsf{S}^k(\mathbb{C}^n)$ be a generic element and let the symmetric outer product decomposition of A be*

$$A = \sum\nolimits_{i=1}^{\overline{R}_{\mathsf{S}}} \mathbf{v}_i^{\otimes k}. \tag{7.2}$$

Then (7.2) *has a finite number of solutions if and only if*

$$\frac{1}{n}\binom{n+k-1}{k} \in \mathbb{N}.$$

TABLE 7.2
Generic dimension $F(k,n)$ of the fiber of solutions.

k \ n	2	3	4	5	6	7	8	9	10
3	0	2	0	**5**	4	0	0	6	0
4	1	**3**	**5**	**5**	0	0	6	0	5
5	0	0	0	4	0	0	0	0	8
6	1	2	0	0	0	0	4	3	5

Actually, one may easily check the generic dimension of the fiber of solutions by computing the number of remaining free parameters [16]:

$$F(k,n) = n\overline{R}_{\mathsf{S}}(k,n) - \binom{n+k-1}{k}.$$

This is summarized in Table 7.2. When the dimension of the fiber is nonzero, there are infinitely many symmetric outer product decompositions.

Our technique is different from the reduction to simplicity proposed by ten Berge [56, 59] but also relies on the calculation of dimensionality.

8. Examples. We will present a few examples to illustrate our discussions in the previous sections.

8.1. Lack of closeness. It has been shown [16, 35] that symmetric tensors of order 3 and dimension 3 have a generic rank $\overline{R}_{\mathsf{S}}(3,3) = 4$ and a maximal rank $R_{\mathsf{S}}(3,3) = 5$. From the results of section 6, this means that only $\mathcal{Z}_4$ is dense in $\overline{\mathcal{Y}}_4 = \overline{\mathcal{Y}}_5$ and that $\mathcal{Z}_3$ and $\mathcal{Z}_5$ are not closed by Proposition 6.5. On the other hand, $\mathcal{Z}_1$ is closed.

To make this statement even more explicit, let us now define a sequence of symmetric tensors, each of symmetric rank 2, that converges to a symmetric tensor of symmetric rank 3. This will be a simple demonstration of the lack of closure of $\mathcal{Y}_r$ for $r > 1$ and $k > 2$, already stated in Proposition 6.8. For this purpose, let $\mathbf{x}, \mathbf{y}$ be two noncollinear vectors. Then the order-3 symmetric tensor

$$A_\varepsilon = \varepsilon^2(\mathbf{x} + \varepsilon^{-1}\mathbf{y})^{\otimes 3} + \varepsilon^2(\mathbf{x} - \varepsilon^{-1}\mathbf{y})^{\otimes 3} \tag{8.1}$$

is of symmetric rank 2 for any scalar $\varepsilon \neq 0$, and it converges, as $\varepsilon \to 0$, to the following symmetric tensor:

$$A_0 = 2\left(\mathbf{x}\otimes\mathbf{y}\otimes\mathbf{y} + \mathbf{y}\otimes\mathbf{x}\otimes\mathbf{y} + \mathbf{y}\otimes\mathbf{y}\otimes\mathbf{x}\right).$$

This limiting symmetric tensor is of symmetric rank 3. In fact, one may show [14] that it admits the following symmetric outer product decomposition:

$$A_0 = (\mathbf{x}+\mathbf{y})^{\otimes 3} - (\mathbf{x}-\mathbf{y})^{\otimes 3} - 2\mathbf{y}^{\otimes 3}.$$

Now let $\mathbf{x}_i, \mathbf{y}_i$ be linearly independent vectors.

By adding two terms of the form (8.1), a similar example can be given in dimension $n = 4$, where we get a sequence of symmetric tensors of symmetric rank 4 converging to a limit of symmetric rank 6.

We will give two more illustrations of Conjecture 6.10.

Example 8.1. If the dimension is $n = 3$, we can take three linearly independent vectors, say, $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$. Then the sequence of symmetric tensors $A_\varepsilon + \mathbf{z}^{\otimes 3}$ is of symmetric rank 3 and converges toward a symmetric rank-4 tensor.

In dimension 3, it is somewhat more tricky to build a sequence converging toward a symmetric tensor of symmetric rank 5. Note that 5 is the maximal rank for $k = 3$ and $n = 3$.

Example 8.2. Consider the sequence below as ε tends to zero:

$$\frac{1}{\varepsilon}\left[(\mathbf{x}+\varepsilon\mathbf{y})^{\otimes 3} - \mathbf{x}^{\otimes 3} + (\mathbf{z}+\varepsilon\mathbf{x})^{\otimes 3} - \mathbf{z}^{\otimes 3}\right]. \tag{8.2}$$

It converges to the following symmetric tensor, which we expressed as a sum of six (asymmetric) rank-1 terms:

$$\mathbf{x}\otimes\mathbf{x}\otimes\mathbf{y} + \mathbf{x}\otimes\mathbf{y}\otimes\mathbf{x} + \mathbf{y}\otimes\mathbf{x}\otimes\mathbf{x} + \mathbf{z}\otimes\mathbf{z}\otimes\mathbf{x} + \mathbf{z}\otimes\mathbf{x}\otimes\mathbf{z} + \mathbf{x}\otimes\mathbf{z}\otimes\mathbf{z}.$$

This has symmetric rank 5 since it can be associated with quantic $x^2y + xz^2$, which is the sum of (at least) five cubes.

In terms of algebraic geometry, this example admits a simple geometric interpretation. The limiting tensor is the sum of a point in the tangent space to $\mathcal{Y}_1$ at $\mathbf{x}^{\otimes 3}$ and a point in the tangent space to $\mathcal{Y}_1$ at $\mathbf{z}^{\otimes 3}$.

Note that the same kind of example can be constructed in the asymmetric case:

$$\mathbf{x}_1\otimes\mathbf{x}_2\otimes(\mathbf{x}_3 - \varepsilon^{-1}\mathbf{y}_3) + (\mathbf{x}_1+\varepsilon\mathbf{y}_1)\otimes(\mathbf{x}_2+\varepsilon\mathbf{y}_2)\otimes\varepsilon^{-1}\mathbf{y}_3.$$

Further discussions of the lack of closeness of $\mathcal{Y}_r$ and the ill-posedness of the best rank-r approximation problem in the asymmetric case can be found in [21].

8.2. Symmetric outer product decomposition over the real field. We now turn our attention to real symmetric tensors. We are interested in the symmetric outer product decomposition of $A \in \mathsf{S}^k(\mathbb{R}^n)$ over $\mathbb{R}$, i.e.,

$$A = \sum\nolimits_{i=1}^{r} \lambda_i \mathbf{v}_i \otimes \mathbf{v}_i \otimes \cdots \otimes \mathbf{v}_i, \tag{8.3}$$

where $\lambda_i \in \mathbb{R}$ and $\mathbf{v}_i \in \mathbb{R}^n$ for all $i = 1, \dots, r$. First note that unlike the decomposition over $\mathbb{C}$ in Lemma 4.2, we can no longer drop the coefficients $\lambda_1, \dots, \lambda_r$ in (8.3) since the kth roots of λ_i may not exist in $\mathbb{R}$.

Since $\mathsf{S}^k(\mathbb{R}^n) \subset \mathsf{S}^k(\mathbb{C}^n)$, we may regard A as an element of $\mathsf{S}^k(\mathbb{C}^n)$ and seek its symmetric outer product decomposition over $\mathbb{C}$. It is easy to see that we will generally need more terms in (8.3) to decompose A over $\mathbb{R}$ than over $\mathbb{C}$ and so

$$\operatorname{rank}_{\mathsf{S},\mathbb{C}}(A) \le \operatorname{rank}_{\mathsf{S},\mathbb{R}}(A). \tag{8.4}$$

This inequality also holds true for the outer product rank of asymmetric tensors. For $k = 2$, i.e., matrices, we always have equality in (8.4), but we will see in the examples below that strict inequality can occur when $k > 2$.

Example 8.3. Let $A \in \mathsf{S}^3(\mathbb{R}^2)$ be defined by

$$A = \left[\begin{array}{cc|cc} -1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right].$$

It is of symmetric rank 3 over $\mathbb{R}$,

$$A = \frac{1}{2}\begin{bmatrix}1\\1\end{bmatrix}^{\otimes 3} + \frac{1}{2}\begin{bmatrix}1\\-1\end{bmatrix}^{\otimes 3} - 2\begin{bmatrix}1\\0\end{bmatrix}^{\otimes 3},$$

whereas it is of symmetric rank 2 over $\mathbb{C}$,

$$A = \frac{\jmath}{2}\begin{bmatrix}-\jmath\\1\end{bmatrix}^{\otimes 3} - \frac{\jmath}{2}\begin{bmatrix}\jmath\\1\end{bmatrix}^{\otimes 3}, \quad \text{where } \jmath := \sqrt{-1}.$$

Hence we see that $\operatorname{rank}_{\mathsf{S},\mathbb{C}}(A) \neq \operatorname{rank}_{\mathsf{S},\mathbb{R}}(A)$.

These decompositions may be obtained using the algorithm described in [16], for instance. Alternatively, this tensor is associated with the homogeneous polynomial in two variables $p(x, y) = 3xy^2 - x^3$, which can be decomposed over $\mathbb{R}$ into

$$p(x, y) = \frac{1}{2}(x + y)^3 + \frac{1}{2}(x - y)^3 - 2x^3.$$

In the case of $2 \times 2 \times 2$ symmetric tensors, or equivalently in the case of binary cubics, the symmetric outer product decomposition can always be computed [16]. Hence, the symmetric rank of any symmetric tensor can be calculated, even over $\mathbb{R}$. In this case, it can be shown that the generic symmetric rank over $\mathbb{C}$ is 2, whereas there are *two typical symmetric ranks* over $\mathbb{R}$, which are 2 and 3.

In fact, in the $2 \times 2 \times 2$ case, there are two 2×2 matrix slices, which we can call A_1 and A_2. Since the generic symmetric rank over $\mathbb{C}$ is 2, the outer product decomposition may be obtained via the eigenvalue decomposition of the matrix pencil (A_1, A_2), which generically exists and whose eigenvalues are those of $A_1 A_2^{-1}$. By generating (four) independent real Gaussian entries with zero mean and unit variance, it can be easily checked out with a simple computer simulation that one gets real eigenvalues in 52% of the cases. This means that the real rank is 3 in 48% of the remaining cases. This is the simplest example demonstrating that a generic rank can be lacking over $\mathbb{R}$. So the concept of typical rank is essential to studying symmetric tensors over $\mathbb{R}$.

For asymmetric tensors, the same kind of computer simulation would yield (by generating eight independent real Gaussian entries) typical ranks of 2 and 3, 78% and 22% of the time, respectively, leading to the same qualitative conclusions. This procedure is not new [54, pp. 13] and was proposed in the past to illustrate the existence of several typical ranks for asymmetric tensors [37, 56]. An interesting result obtained by ten Berge [57] is that $p \times p \times 2$ real asymmetric tensors have typical ranks $\{p, p + 1\}$.

The problems pertaining to rank and decompositions of real symmetric tensors have not received as much attention as their complex counterparts. However, a moderate amount of work has been done [37, 48, 56, 59, 58], and we refer the reader to these for further information.

8.3. Open questions. Most of the results that we have presented so far are limited to symmetric tensors over the complex field. The case of general asymmetric tensors is currently being addressed with the same kind of approach. As pointed out earlier, decompositions over the real field are more complicated to handle with algebraic geometric tools. In addition, while the problem of determining the generic symmetric rank has been resolved thanks to the Alexander–Hirschowitz theorem, the maximal symmetric rank is known only for particular values of order and dimensions (e.g., dimension 2); only very rough upper bounds are known for general values. Lastly, the computation of an explicit symmetric outer product decomposition for a symmetric tensor is computationally expensive, and the conditions (dimension, order) under which this can be executed within a polynomial time are not yet clearly known. These are problems that we hope will be addressed in future work, either by ourselves or by interested readers.

Acknowledgments. The authors would like to thank the anonymous reviewers for their helpful comments. This work is a result of collaboration initiated at the 2004 Workshop on Tensor Decomposition held at the American Institute of Mathematics, Palo Alto, California, and continued in the 2005 Workshop on Tensor Decomposition

and Its Applications held at Centre International de Rencontres Mathématiques, Luminy, France.

REFERENCES

[1] J. Alexander and A. Hirschowitz, *Polynomial interpolation in several variables*, J. Algebraic Geom., 4 (1995), pp. 201–222.

[2] J. Alexander and A. Hirschowitz, *La méthode d'Horace éclatée: application à l'interpolation en degré quatre*, Invent. Math., 107 (1992), pp. 585–602.

[3] M. D. Atkinson and S. Lloyd, *Bounds on the ranks of some* 3*-tensors*, Linear Algebra Appl., 31 (1980), pp. 19–31.

[4] D. Bini, M. Capovani, G. Lotti, and F. Romani, $O(n^{2.7799})$ *complexity for* $n \times n$ *approximate matrix multiplication*, Inform. Process. Lett., 8 (1979), pp. 234–235.

[5] N. Bourbaki, *Algebra* I, Elements of Mathematics, Springer-Verlag, Berlin, 1998.

[6] D. R. Brillinger, *Time Series: Data Analysis and Theory*, Classics in Appl. Math. 36, SIAM, Philadelphia, 2001.

[7] A. Brini, R. Q. Huang, and A. G. B. Teolis, *The umbral symbolic method for supersymmetric tensors*, Adv. Math., 96 (1992), pp. 123–193.

[8] J. W. Brewer, *Kronecker products and matrix calculus in system theory*, IEEE Trans. Circuits Systems, 25 (1978), pp. 772–781.

[9] P. Burgisser, M. Clausen, and M. A. Shokrollahi, *Algebraic Complexity Theory*, 315, Springer-Verlag, Berlin, 1997.

[10] J. F. Cardoso, *Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors*, in Proceedings of the IEEE International Conference on Acoustical Speech Signal Processing (ICASSP), Toronto, 1991, pp. 3109–3112.

[11] J. D. Carroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart-Young decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[12] K. Chandler, *Linear systems of cubics singular at general points of projective space*, Compos. Math., 134 (2002), pp. 269–282.

[13] G. Comas and M. Seiguer, *On the Rank of a Binary Form*, preprint, arXiv:math/0112311v1, 2001.

[14] P. Comon, *Tensor decompositions*, in Mathematics in Signal Processing V, J. G. McWhirter and I. K. Proudler, eds., Clarendon Press, Oxford, UK, 2002, pp. 1–24.

[15] P. Comon, *Blind identification and source separation in* 2×3 *under-determined mixtures*, IEEE Trans. Signal Process., 52 (2004), pp. 11–22.

[16] P. Comon and B. Mourrain, *Decomposition of quantics in sums of powers of linear forms*, Signal Process., 53 (1996), pp. 93–107.

[17] P. Comon and M. Rajih, *Blind identification of under-determined mixtures based on the characteristic function*, Signal Process., 86 (2006), pp. 2271–2281.

[18] D. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*, Graduate Texts in Math. 185, Springer-Verlag, New York, 1998.

[19] L. de Lathauwer, *A Link Between Canonical Decomposition in Multilinear Algebra and Simultaneous Matrix Diagonalization*, preprint, 2006.

[20] L. de Lathauwer, B. de Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[21] V. de Silva and L.-H. Lim, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1084–1127.

[22] R. Ehrenborg and G. C. Rota, *Apolarity and canonical forms for homogeneous polynomials*, European J. Combin., 14 (1993), pp. 157–181.

[23] D. Eisenbud, *Lectures on the Geometry of Syzygies*, Math. Sci. Res. Inst. Publ. 51, Cambridge University Press, Cambridge, UK, 2004, pp. 115–152.

[24] L. Eldén and B. Savas, *A Newton–Grassmann Method for Computing the Best Multilinear rank-*(r_1, r_2, r_3) *Approximation of a Tensor*, preprint, 2007.

[25] D. S. Freed, *Five Lectures on Supersymmetry*, AMS, Providence, RI, 1999.

[26] W. Greub, *Multilinear Algebra*, 2nd ed., Springer-Verlag, New York, 1978.

[27] J. Harris, *Algebraic Geometry: A First Course*, Graduate Texts in Math. 133, Springer-Verlag, New York, 1998.

[28] R. A. Harshman, *Foundations of the* PARAFAC *procedure: Models and conditions for an explanatory multimodal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[29] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp. 164–189.
[30] F. L. Hitchcock, *Multiple invariants and generalized rank of a p-way matrix or tensor*, J. Math. Phys., 7 (1927), pp. 39–79.
[31] T. D. Howell, *Global properties of tensor rank*, Linear Algebra Appl., 22 (1978), pp. 9–23.
[32] J. JáJá, *An addendum to Kronecker's theory of pencils*, SIAM J. Appl. Math., 37 (1979), pp. 700–712.
[33] A. M. Kagan, Y. V. Linnik, and C. R. Rao, *Characterization Problems in Mathematical Statistics*, John Wiley, New York, 1973.
[34] E. Kofidis and P. Regalia, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2001/02), pp. 863–884.
[35] I. A. Kogan and M. M. Maza, *Computation of canonical forms for ternary cubics*, Proceedings of the International Symposium Symbolic Algebraic Computation (ISAAC), ACM Press, New York, 2002, pp. 151–160.
[36] J. B. Kruskal, *Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.
[37] J. B. Kruskal, *Rank, decomposition, and uniqueness for 3-way and n-way arrays*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North–Holland, Amsterdam, 1989, pp. 7–18.
[38] J. Kung and G. Rota, *The invariant theory of binary forms*, Bull. Amer. Math. Soc., 10 (1984), pp. 27–85.
[39] S. Lang, *Algebra*, 3rd ed., Graduate Texts in Math. 211, Springer-Verlag, New York, 2002.
[40] S. Lang, *Real and Functional Analysis*, 3rd ed., Graduate Texts in Math. 142, Springer-Verlag, New York, 1993.
[41] L.-H. Lim, *Optimal solutions to non-negative* parafac*/multilinear* nmf *always exist*, Workshop on Tensor Decompositions and Applications, Centre International de rencontres Mathématiques, Luminy, France, 2005.
[42] M. Marcus, *Finite Dimensional Multilinear Algebra*, Part I, Monogr. Textbooks Pure Appl. Math. 23, Marcel Dekker, New York, 1973.
[43] M. Marcus, *Finite Dimensional Multilinear Algebra*, Part II, Monogr. Textbooks Pure Appl. Math. 23, Marcel Dekker, New York, 1975.
[44] P. McCullagh, *Tensor Methods in Statistics*, Chapman and Hall, London, 1987.
[45] D. G. Northcott, *Multilinear Algebra*, Cambridge University Press, Cambridge, UK, 1984.
[46] L. Qi, *Eigenvalues of a real supersymmetric tensor*, J. Symbolic Comput., 40 (2005), pp. 1302–1324.
[47] B. Reznick, *Sums of Powers of Complex Linear Forms*, private correspondence, August 1992.
[48] B. Reznick, *Sums of even powers of real linear forms*, Mem. Amer. Math. Soc., 96 (463) (1992).
[49] I. R. Shafarevitch, *Basic Algebraic Geometry*, Grundlehren Math. Wiss. 213, Springer-Verlag, Berlin, 1977.
[50] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, *Parallel factor analysis in sensor array processing*, IEEE Trans. Signal Process., 48 (2000), pp. 2377–2388.
[51] A. Smilde, R. Bro, and P. Geladi, *Multi-Way Analysis*, John Wiley, West Sussex, UK, 2004.
[52] V. Strassen, *Rank and optimal computation of generic tensors*, Linear Algebra Appl., 52/53 (1983), pp. 645–685.
[53] J. J. Sylvester, *Sur une extension d'un théorème de Clebsch relatif aux courbes du quatrième degré*, C. R. Math. Acad. Sci. Paris, 102 (1886), pp. 1532–1534.
[54] J. M. F. ten Berge, *Partial uniqueness in* candecomp*/*parafac, J. Chemometrics, 18 (2004), pp. 12–16.
[55] J. M. F. ten Berge, H. A. L. Kiers, and W. P. Krijnen, *Computational solutions for the problem of negative saliences and nonsymmetry in* indscal, J. Classification, 10 (1993), pp. 115–124.
[56] J. M. F. ten Berge and H. A. L. Kiers, *Simplicity of core arrays in three-way principal component analysis and the typical rank of $p \times q \times 2$ arrays*, Linear Algebra Appl., 294 (1999), pp. 169–179.
[57] J. M. F. ten Berge, *The typical rank of tall three-way arrays*, Psychometrika, 65 (2000), pp. 525–532.
[58] J. M. F. ten Berge, N. D. Sidiropoulos, and R. Rocci, *Typical rank and* indscal *dimensionality for symmetric three-way arrays of order $I \times 2 \times 2$ or $I \times 3 \times 3$*, Linear Algebra Appl., 388 (2004), pp. 363–377.
[59] J. M. F. ten Berge, *Simplicity and typical rank of three-way arrays, with applications to Tucker-3 analysis with simple cores*, J. Chemometrics, 18 (2004), pp. 17–21.

[60] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
[61] V. S. Varadarajan, *Supersymmetry for Mathematicians: An Introduction*, Courant Lecture Notes Math. 11, AMS, Providence, RI, 2004.
[62] D. A. Weinstein, *Canonical forms for symmetric tensors*, Linear Algebra Appl., 57 (1984), pp. 271–282.
[63] P. West, *Introduction to Supersymmetry and Supergravity*, 2nd ed., World Scientific, Teaneck, NJ, 1990.
[64] T. Yokonuma, *Tensor Spaces and Exterior Algebra*, Transl. Math. Monogr. 108, AMS, Providence, RI, 1992.
[65] F. L. Zak, *Tangents and Secants of Algebraic Varieties*, Transl. Math. Monogr. 127, AMS, Providence, RI, 1993.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1281–1301

TOWARDS A BACKWARD PERTURBATION ANALYSIS FOR DATA LEAST SQUARES PROBLEMS*

X.-W. CHANG†, G. H. GOLUB‡, AND C. C. PAIGE†

Abstract. Given an approximate solution to a data least squares (DLS) problem, we would like to know its minimal backward error. Here we derive formulas for what we call an "extended" minimal backward error, which is at worst a lower bound on the minimal backward error. When the given approximate solution is a good enough approximation to the exact solution of the DLS problem (which is the aim in practice), the extended minimal backward error is the actual minimal backward error, and this is also true in other easily assessed and common cases. Since it is computationally expensive to compute the extended minimal backward error directly, we derive a lower bound on it and an asymptotic estimate for it, both of which can be evaluated less expensively. Simulation results show that for reasonable approximate solutions, the lower bound has the same order as the extended minimal backward error, and the asymptotic estimate is an excellent approximation to the extended minimal backward error.

Key words. data least squares, backward errors, numerical stability, perturbation analysis, asymptotic estimate, iterative methods, stopping criteria

AMS subject classifications. 15A06, 65F20, 65G50

DOI. 10.1137/060668626

1. Introduction.

Given an approximate solution to a problem, the aim of backward perturbation analysis is to find a minimum size perturbation in the data such that the approximate solution is an exact solution of the perturbed problem. In the analysis one tries to find a formula for, or good bounds on, the size of the minimal perturbation (to be referred to as the minimal backward error) and design an efficient algorithm to evaluate or estimate the formula or the bounds. If the relative minimal backward error (i.e., the size of the minimal perturbation divided by an acceptable measure of the size of the data) is of the order of the unit roundoff, then we say that the approximate solution is a (normwise) backward stable solution. Backward perturbation analyses are useful in practice. Sometimes we may not know if an algorithm for solving a problem is numerically stable, e.g., the backward numerical stability of some fast algorithms for structured matrix problems is unknown. But if we know that a computed solution of a specific problem is a backward stable solution, we are satisfied with this computed solution. Also when we solve a large-scale problem by an iterative algorithm, the results of a backward perturbation analysis can often be used to design effective stopping criteria; see, for example, [1], [20], and [25].

There has been a lot of work on the backward perturbation analysis of linear systems, especially in recent years. For example, for consistent linear systems, see [14], [25], [30], [31], [32], [34], [36]; for unconstrained least squares problems, see [9],

*Received by the editors August 30, 2006; accepted for publication (in revised form) by N. Mastronardi March 31, 2008; published electronically October 16, 2008.

http://www.siam.org/journals/simax/30-4/66862.html

†School of Computer Science, McGill University, Montreal, Quebec, H3A 2A7 Canada, (chang@cs.mcgill.ca, paige@cs.mcgill.ca). The work of the first author was supported by NSERC of Canada grant RGPIN217191-03. The third author's work was supported by NSERC of Canada grant RGPIN9236.

‡This author is deceased. Former address: Department of Computer Science, Stanford University, Stanford, CA 94305-9025. This author's work was in part supported by NSF grant QAACT.

[12], [17], [18], [19], [26], [27], [28], [29], [30], [35]; and for constrained least squares problems, see [4], [18], and [19].

The main purpose of this paper is to give a normwise backward perturbation analysis for the general linear data least squares (DLS) problem. As a result, the structure of the matrix and magnitudes of individual elements of the matrix in the DLS problem will not be considered. We derive formulas for an "extended" minimal backward error in section 2. This extended minimal backward error is at worst a lower bound on the minimal backward error. But we show that when the given approximate solution is a good enough approximation to the exact solution of the DLS problem (which is the aim in practice), the extended minimal backward error is the actual minimal backward error. Section 2.1 deals with perturbations in both A and b, while section 2.2 considers perturbations in A alone and shows how these are limiting cases of those in section 2.1. Since computing the extended minimal backward error directly is time consuming, in section 3 we derive a lower bound on, and in section 4 an asymptotic estimate for, this extended minimal backward error. We give numerical examples in section 5. Finally a summary is given in section 6.

We use $I=[e_1,\ldots,e_n]$ to denote the unit matrix. For any matrix $B\in\mathbb{R}^{m\times n}$, its column range is denoted by $\mathcal{R}(B)$, its Moore–Penrose generalized inverse is denoted by $B^\dagger$, its smallest singular value (the pth largest singular value, with $p=\min\{m,n\}$) by $\sigma_{\min}(B)$, and its condition number in the 2-norm is denoted by $\kappa_2(B)$. For any matrix $B=[b_1,\ldots,b_n]$ $\mathtt{vec}(B)=[b_1^T,\ldots,b_n^T]^T$. For any symmetric $B\in\mathbb{R}^{n\times n}$, its eigenvalues are labeled in nondecreasing order: $\lambda_{\min}=\lambda_1\le\lambda_2\le\cdots\le\lambda_n$, but when only $\lambda_{\min}$ is of interest we will write $\lambda=\lambda_{\min}$. For any vector $v\in\mathbb{R}^n$, its Moore–Penrose generalized inverse is

$$v^\dagger\equiv\begin{cases}0 & \text{if } v=0,\\ v^T/\|v\|_2^2 & \text{if } v\neq 0,\end{cases}\qquad \|v\|_2\equiv(v^Tv)^{\frac{1}{2}}.$$

Note that $vv^\dagger$ is the orthogonal projector onto $\mathcal{R}(v)$, and $I-vv^\dagger$ is the orthogonal projector onto the orthogonal complement of $\mathcal{R}(v)$.

2. Backward perturbation analysis. Given $A\in\mathbb{R}^{m\times n}$ and $b\in\mathbb{R}^m$, the DLS problem defined by DeGroat and Dowling [5] is

$$\sigma_{\scriptscriptstyle D}\equiv\min_{E,x}\|E\|_F\quad\text{subject to}\quad(A+E)x=b,\qquad \|E\|_F\equiv\left[\operatorname{trace}\left(E^TE\right)\right]^{\frac{1}{2}}. \tag{2.1}$$

See also, for example, [21], [22]. The purpose of the DLS problem is to find the optimal x. For applications of the DLS method to some signal processing problems, see [5]. Let $\mathcal{U}_{\min}(A)$ be the left singular vector subspace of A corresponding to its minimum singular value $\sigma_{\min}(A)$. In [21] it was explained that a satisfactory condition for building the theory for the DLS problem (2.1) is the condition that we will now assume holds:

$$A \text{ has full column rank, and } b\not\perp\mathcal{U}_{\min}(A). \tag{2.2}$$

With this condition, the solution to (2.1) must exist and be unique. From $(A+E)x=b$ we have $Ex=b-Ax$. Thus the minimal E must satisfy

$$E=(b-Ax)x^\dagger. \tag{2.3}$$

But (2.2) implies that $b\neq 0$, so the solution x must be nonzero, and this allows us to eliminate E and reformulate the DLS problem (2.1) as

$$\sigma_{\scriptscriptstyle D}\equiv\min_x\|(b-Ax)x^\dagger\|_F=\min_x\frac{\|b-Ax\|_2}{\|x\|_2}. \tag{2.4}$$

From [21, equations (5.14)–(5.17)], $\hat{x}$ solves the DLS problem (2.1) if and only if

$$A^T(b - A\hat{x}) = -\hat{x}\frac{\|b - A\hat{x}\|_2^2}{\|\hat{x}\|_2^2}, \tag{2.5}$$

$$\frac{\|b - A\hat{x}\|_2}{\|\hat{x}\|_2} < \sigma_{\min}(A). \tag{2.6}$$

Differentiating the objective function in (2.4) and setting the result to zero gives (2.5), corresponding to a stationary point. The global minimum also satisfies (2.6).

The DLS formulation is designed for problems where the right-hand side b is accurately known, but the matrix A is only known approximately. Given a nonzero approximate solution $y \in \mathbb{R}^n$ to $Ax \approx b$, two questions are of particular interest here:

Q1: Is y a feasible (not necessarily DLS) solution, given the accuracy of the data?

Q2: Is y a backward stable solution to the DLS problem for the given data A, b?

Q1 will often be easy to check: for example, if it is known that the given data matrix A approximates an unknown ideal matrix $\hat{A}$ to within $\|A - \hat{A}\|_{\{F \text{ or } 2\}} \le \alpha$ while b is accurately known, then from (2.3) we need only check that $\|b - Ay\|/\|y\| \le \alpha$. If the answer to Q1 is positive and we are not interested in the DLS solution, we might accept y. But then in practice there will be an infinite set of y satisfying Q1, and we will often seek some additional criterion, for example, "Does y make sense physically?"—a difficult question we might ask of an ill-posed problem. Here we consider the more generally approachable question Q2, since if we can answer this affirmatively, we will know that y is a desirable [illegible] solution to (2.1). Even if the answer to Q1 is "no," we might still check Q2, since it is possible for y to satisfy Q2 but not Q1 in that y can be a DLS solution for $A + \Delta A$, $b + \Delta b$ for very small ΔA and Δb, but the minimal norm E in $(A + \Delta A + E)y = b + \Delta b$ can be too large for Q1. This would indicate that there are difficulties with the data.

To answer Q2 we would like to solve the minimal backward error problem:

$$\min_{\Delta A, \Delta b} \|[\Delta A, \Delta b\, \theta]\|_F \quad \text{subject to} \quad y = \arg\min_x \frac{\|b + \Delta b - (A + \Delta A)x\|_2}{\|x\|_2}; \tag{2.7}$$

see (2.4), where the chosen scalar $\theta \ge 0$ allows a different emphasis on each data error.

From (2.5) and (2.6) we see that $[\Delta A, \Delta b]$ is a backward perturbation for the DLS problem with the given solution y if and only if it is in the set $\mathcal{C}_{A,b}$, where

$$\mathcal{C}^+_{A,b} \equiv \left\{ [\Delta A, \Delta b] : \ (A+\Delta A)^T[b+\Delta b - (A+\Delta A)y] = -y\frac{\|b+\Delta b - (A+\Delta A)y\|_2^2}{\|y\|_2^2} \right\}, \tag{2.8}$$

$$\mathcal{C}_{A,b} \equiv \left\{ [\Delta A, \Delta b] : \ [\Delta A, \Delta b] \in \mathcal{C}^+_{A,b} \ \& \ \frac{\|b+\Delta b - (A+\Delta A)y\|_2}{\|y\|_2} < \sigma_{\min}(A+\Delta A) \right\}. \tag{2.9}$$

The inequality in (2.9) makes it difficult to derive a general expression for $[\Delta A, \Delta b] \in \mathcal{C}_{A,b}$, so we initially ignore it and consider the larger set $\mathcal{C}^+_{A,b}$, which we will show is also useful. The following result from Theorem 5.1 of [3] characterizes $[\Delta A, \Delta b] \in \mathcal{C}^+_{A,b}$.

LEMMA 2.1. [illegible] $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, [illegible] $y \in \mathbb{R}^n$ [illegible] $[\Delta A, \Delta b] \in \mathcal{C}^+_{A,b}$ [illegible] (2.8) [illegible] $w \in \mathbb{R}^n$ [illegible] $Z \in \mathbb{R}^{m\times n}$ [illegible]

$$A + \Delta A = (b + \Delta b - w)y^\dagger + \left(I - ww^\dagger\right) Z \left(I - yy^\dagger\right), \quad (b + \Delta b)^T w = 0. \tag{2.10}$$

2.1. Allowing backward perturbations in A and b. Based on Lemma 2.1 we will first find a computable expression for $\mu_F(y,\theta)$ in the following "extended" minimal backward error problem:

$$\mu_F(y,\theta) \equiv \min_{[\Delta A,\Delta b]\in\mathcal{C}^+_{A,b}} \|[\Delta A, \Delta b\,\theta]\|_F \quad \text{for } \mathcal{C}^+_{A,b} \text{ in (2.8).} \tag{2.11}$$

We call $\mu_F(y,\theta)$ the extended minimal backward error because we minimize over the extended set $\mathcal{C}^+_{A,b}$, giving at worst a lower bound on the minimal backward error.

If $b=0$, the DLS problem (2.1) has the solution $\hat{x}=0$; if $y=0$, then it cannot be the DLS solution of any problem with $b\neq 0$. We do not need to consider these cases further. For the remainder of this paper we will assume the conditions and notation of the following theorem.

The following will simplify the presentation:

$$\rho \equiv 1/\left(1+\theta^2\|y\|_2^2\right) \quad \text{so that} \quad \rho\theta^2\|y\|_2^2 = 1-\rho \quad \text{and} \quad 0\le\rho\le 1. \tag{2.12}$$

THEOREM 2.2. [illegible] $A\in\mathbb{R}^{m\times n}$, $b\in\mathbb{R}^m$, [illegible] $y\in\mathbb{R}^n$, $\theta\ge 0$ [illegible] (2.2) [illegible] $r\equiv b-Ay$, $\rho\equiv 1/(1+\theta^2\|y\|_2^2)$ [illegible]

$$N \equiv N(\theta) \equiv \left[A\left(I-yy^\dagger\right), \rho^{\frac{1}{2}}\theta\|r\|_2\left(I-rr^\dagger\right), b\,\theta\right], \tag{2.13}$$

$$M \equiv M(\theta) \equiv A\left(I-yy^\dagger\right)A^T - r\rho\theta^2 r^T + b\,\theta^2 b^T = NN^T - \rho\theta^2\|r\|_2^2 I. \tag{2.14}$$

[illegible] $M(\theta)$ [illegible] $\mu_F(y,\theta)$ [illegible] (2.11) [illegible]

$$\mu_F^2(y,\theta) = \begin{cases} \rho\theta^2\|r\|_2^2 & \text{if } \lambda_{\min}(M(\theta))\ge 0, \\ \rho\theta^2\|r\|_2^2 + \lambda_{\min}(M(\theta)) = \sigma_{\min}^2(N(\theta)) & \text{if } \lambda_{\min}(M(\theta))<0. \end{cases} \tag{2.15}$$

[illegible] $\mu_F(y,\theta)$ [illegible] $\widehat{\Delta A}$ [illegible] $\widehat{\Delta b}$.

$$A+\widehat{\Delta A} = \begin{cases} A+r(1-\rho)y^\dagger & \text{if } \lambda_{\min}(M(\theta))\ge 0, \\ \left(I-w_\theta w_\theta^\dagger\right)[A+r(1-\rho)y^\dagger]+w_\theta w_\theta^\dagger Ayy^\dagger & \text{if } \lambda_{\min}(M(\theta))<0, \end{cases} \tag{2.16}$$

$$b+\widehat{\Delta b} = \begin{cases} b-r\rho & \text{if } \lambda_{\min}(M(\theta))\ge 0, \\ \left(I-w_\theta w_\theta^\dagger\right)(b-r\rho) & \text{if } \lambda_{\min}(M(\theta))<0, \end{cases} \tag{2.17}$$

[illegible] w_θ [illegible] ($\lambda_{\min}(M(\theta))<0$) [illegible] $M(\theta)$ [illegible] $\lambda_{\min}(M(\theta))$ [illegible] $N(\theta)$ [illegible] $\sigma_{\min}(N(\theta))$ [illegible] $A(I-yy^\dagger)A^T+b\,\theta^2 b^T$ is nonnegative definite, so from [15, Theorem 4.3.4(b)] with $k=1$, $M(\theta)$ has at most one negative eigenvalue.

Now we want to determine the optimal w, Z, ΔA, and Δb in (2.10) to minimize $\|\Delta A, \Delta b\,\theta\|_F$. In the following we discuss two cases separately.

[illegible] 1: The optimal $w=0$. Let $Y=[y/\|y\|_2, Y_2]\in\mathbb{R}^{n\times n}$ be an orthogonal matrix. From (2.10) we have $(b+\Delta b)^T w=0$ automatically, and

$$\begin{aligned} \Delta A Y &= (b+\Delta b)y^\dagger[y/\|y\|_2, Y_2] + Z\left(I-yy^\dagger\right)[y/\|y\|_2, Y_2] - A[y/\|y\|_2, Y_2] \\ &= [(b+\Delta b)/\|y\|_2, 0] + [0, ZY_2] - [Ay/\|y\|_2, AY_2] \\ &= [(r+\Delta b)/\|y\|_2, (Z-A)Y_2]. \end{aligned}$$

It follows that

$$\|[\Delta A, \Delta b\,\theta]\|_F^2 = \|\Delta A Y\|_F^2 + \theta^2\|\Delta b\|_2^2 = \frac{1}{\|y\|_2^2}\left\|\begin{bmatrix} I \\ \theta\|y\| I\end{bmatrix}\Delta b + \begin{bmatrix} r \\ 0\end{bmatrix}\right\|_2^2 + \|(Z-A)Y_2\|_F^2.$$

Thus $\|[\Delta A, \Delta b\,\theta]\|_F$ is minimized when

$$\Delta b = \widehat{\Delta b} \equiv -\begin{bmatrix} I \\ \theta\|y\| I\end{bmatrix}^{\dagger}\begin{bmatrix} r \\ 0\end{bmatrix} = -r\rho, \quad Z = \widehat{Z} \equiv A, \tag{2.18}$$

and from (2.10) we see that the optimal ΔA must satisfy

$$\Delta A = \widehat{\Delta A} \equiv (b - r\rho)y^{\dagger} + A\left(I - yy^{\dagger}\right) - A = r(1-\rho)y^{\dagger}, \tag{2.19}$$

$$\|[\widehat{\Delta A}, \widehat{\Delta b}\,\theta]\|_F^2 = (1-\rho)^2\|ry^{\dagger}\|_2^2 + \rho^2\theta^2\|r\|_2^2 = \rho\theta^2\|r\|_2^2. \tag{2.20}$$

In Case 2 we will show that if $\lambda_{\min}(M) \geq 0$, then $w = 0$ is optimal.

2: The optimal $w \neq 0$. Let Y be as in Case 1, and $W = [w/\|w\|_2, W_2] \in \mathbb{R}^{m\times m}$ be an orthogonal matrix. Since $w^T(b + \Delta b) = 0$, we can write

$$b + \Delta b = W_2 s \quad \text{for some } s \in \mathbb{R}^{m-1}. \tag{2.21}$$

From (2.10) we have

$$\begin{aligned} W^T\Delta A Y &= \begin{bmatrix} w^T/\|w\|_2 \\ W_2^T\end{bmatrix}\left[(b + \Delta b - w)y^{\dagger} + \left(I - ww^{\dagger}\right)Z\left(I - yy^{\dagger}\right) - A\right][y/\|y\|_2, Y_2] \\ &= \left[\begin{array}{c|c} -\|w\|_2/\|y\|_2 - w^TAy/(\|w\|_2\|y\|_2) & -w^TAY_2/\|w\|_2 \\ \hline s/\|y\|_2 - W_2^TAy/\|y\|_2 & W_2^TZY_2 - W_2^TAY_2\end{array}\right]. \end{aligned}$$

Thus the objective function can be written as five additive nonnegative terms:

$$\begin{aligned} \|[\Delta A, \Delta b\,\theta]\|_F^2 = &\left[\|w\|_2/\|y\|_2 + w^TAy/(\|w\|_2\|y\|_2)\right]^2 + \|w^TAY_2\|_2^2/\|w\|_2^2 \\ &+ \|s - W_2^TAy\|_2^2/\|y\|_2^2 + \|W_2^T(Z-A)Y_2\|_F^2 + \theta^2\|W_2 s - b\|_2^2. \end{aligned} \tag{2.22}$$

To minimize this we take $Z = \widehat{Z} \equiv A$ and note the sum of terms involving s is

$$\begin{aligned} \phi(s) &\equiv \|s - W_2^TAy\|_2^2/\|y\|_2^2 + \theta^2\|W_2 s - b\|_2^2 \\ &= \left[\left\|W^T(W_2 s - Ay)\right\|_2^2 - \left(w^TAy\right)^2/\|w\|_2^2 + \theta^2\|y\|_2^2\|W_2 s - b\|_2^2\right]/\|y\|_2^2 \\ &= \frac{1}{\|y\|_2^2}\left\|\begin{bmatrix} I \\ \theta\|y\|_2 I\end{bmatrix}W_2 s - \begin{bmatrix} Ay \\ b\,\theta\|y\|_2\end{bmatrix}\right\|_2^2 - \left(\frac{w^TAy}{\|w\|_2\|y\|_2}\right)^2. \end{aligned} \tag{2.23}$$

The normal equations for $\hat{s}$, the optimal s, give $(1 + \theta^2\|y\|_2^2)\hat{s} = W_2^T(Ay + b\,\theta^2\|y\|_2^2)$. Therefore

$$\hat{s} = W_2^T[Ay\rho + b(1-\rho)] = W_2^T(b - r\rho). \tag{2.24}$$

Substituting this in the first line of (2.23) gives with $W_2W_2^T = I - ww^{\dagger}$

$$\begin{aligned} \phi(\hat{s}) &= \left\|W_2^T r(1-\rho)\right\|_2^2/\|y\|_2^2 + \theta^2\left\|ww^{\dagger}b + W_2W_2^T r\rho\right\|_2^2 \\ &= \rho\theta^2\left\|\left(I - ww^{\dagger}\right)r\right\|_2^2 + \theta^2\left\|ww^{\dagger}b\right\|_2^2. \end{aligned}$$

Then from (2.22), we obtain

$$\min_{[\Delta A,\Delta b]\in\mathcal{C}^+_{A,b}} \|[\Delta A, \Delta b\,\theta]\|_F^2 = \min_w [\psi_1(w) + \psi_2(w)], \tag{2.25}$$

$$\psi_1(w) \equiv \left[\|w\|_2/\|y\|_2 + w^T Ay/(\|w\|_2\|y\|_2)\right]^2, \tag{2.26}$$

$$\psi_2(w) \equiv \left\|w^T AY_2\right\|_2^2/\|w\|_2^2 + \rho\theta^2 \left\|(I - ww^\dagger)r\right\|_2^2 + \theta^2 \left\|ww^\dagger b\right\|_2^2. \tag{2.27}$$

We will minimize $\psi_2(w)$, which is a function of $w/\|w\|_2$ alone, and then show that we can set $\psi_1(w)$ to zero by scaling w, leading to the optimal w. Since $Y_2Y_2^T = I - yy^\dagger$,

$$\begin{aligned}\psi_2(w) &= \frac{w^T A\left(I - yy^\dagger\right)A^T w}{w^T w} + \rho\theta^2\|r\|_2^2\frac{w^T\left(I - rr^\dagger\right)w}{w^T w} + \theta^2\frac{w^T bb^T w}{w^T w} \\ &= \frac{w^T NN^T w}{w^T w} = \rho\theta^2\|r\|_2^2 + \frac{w^T Mw}{w^T w},\end{aligned} \tag{2.28}$$

whose minimum is $\rho\theta^2\|r\|_2^2 + \lambda_{\min}(M)$ given by $w = w_\theta\alpha$ for any nonzero $\alpha \in \mathbb{R}$ and w_θ satisfying $Mw_\theta = w_\theta\lambda_{\min}(M)$, $\|w_\theta\|_2 = 1$, since we assumed $w \neq 0$.

If $\lambda_{\min}(M) \geq 0$, the above with (2.20) in Case 1 show that $w = 0$ is optimal for minimizing $\|[\Delta A, \Delta b\,\theta]\|_F^2$, giving the minimum value $\rho\theta^2\|r\|_2^2$. So from (2.18), (2.19), and (2.20) we see that the top equalities in each of (2.15), (2.16), and (2.17) hold. Only when $\lambda_{\min}(M) < 0$ do we need to consider the possibility that $w \neq 0$.

Assume that $\lambda_{\min}(M) < 0$. It is easy to verify that $\psi_1(\hat{w}) = 0$ if

$$\hat{w} \equiv -w_\theta\left(w_\theta^T Ay\right) \neq 0. \tag{2.29}$$

Suppose $w_\theta^T Ay = 0$, then from $w_\theta^T Mw_\theta = \lambda_{\min}(M) < 0$ and (2.14),

$$0 > \lambda_{\min}(M) = w_\theta^T AA^T w_\theta - \left(w_\theta^T r\right)^2\rho\theta^2 + \left(w_\theta^T b\right)^2\theta^2 = w_\theta^T AA^T w_\theta + \left(w_\theta^T b\right)^2(1-\rho)\theta^2,$$

which is impossible since the right-hand side is nonnegative; see (2.12), proving that the inequality in (2.29) holds. Therefore from (2.25) we see that when $\lambda_{\min}(M) < 0$, the extended minimal backward error $\mu_F(y,\theta)$ satisfies the two bottom equalities in (2.15). The bottom equality in (2.17) follows immediately from (2.21) and (2.24), and substituting this with $Z = A$ and (2.29) in (2.10) gives

$$\begin{aligned}A + \Delta A &= \left[\left(I - w_\theta w_\theta^\dagger\right)(b - r\rho) + w_\theta\left(w_\theta^T Ay\right)\right]y^\dagger + \left(I - w_\theta w_\theta^\dagger\right)A\left(I - yy^\dagger\right) \\ &= \left(I - w_\theta w_\theta^\dagger\right)\left[A + (b - r\rho - Ay)y^\dagger\right] + w_\theta\left(w_\theta^T Ay\right)y^\dagger \\ &= \left(I - w_\theta w_\theta^\dagger\right)\left[A + r(1-\rho)y^\dagger\right] + w_\theta\left(w_\theta^T Ay\right)y^\dagger\end{aligned}$$

to prove the bottom equation in (2.16). ∎

2.1. The criterion $\lambda_{\min}(M(\theta)) \geq 0$ appears in (2.15)–(2.17). But if, as is usual, $m > n+1$, then $M(\theta)$ has at least $m-n-1$ zero eigenvalues corresponding to eigenvectors spanning $\mathcal{R}([A,b])^\perp$. The eigenvalues of a parameterized matrix that are zero independent of the parameter (here θ) will be called "trivial zero eigenvalues." Because they remain zero, their limiting behavior is trivial.

2.2. Allowing a backward perturbation in *A* alone. In DLS problems only the matrix A is assumed to have uncertainty, so it is also important to consider the

case where there is a backward perturbation in A alone. Then the corresponding minimal backward error problem becomes

$$\min_{\Delta A\in\mathcal{C}_A}\|\Delta A\|_F, \qquad \text{where}$$

$$\mathcal{C}_A^+ \equiv \left\{\Delta A : (A+\Delta A)^T[b-(A+\Delta A)y] = -y\frac{\|b-(A+\Delta A)y\|_2^2}{\|y\|_2^2}\right\}, \tag{2.30}$$

$$\mathcal{C}_A \equiv \left\{\Delta A :\ \Delta A\in\mathcal{C}_A^+ \ \&\ \frac{\|b-(A+\Delta A)y\|_2}{\|y\|_2} < \sigma_{\min}(A+\Delta A)\right\}, \tag{2.31}$$

and these two sets are just $\mathcal{C}_{A,b}^+$ and $\mathcal{C}_{A,b}$ in (2.8) and (2.9) with $\Delta b = 0$, so that $\Delta A\in\mathcal{C}_A^+ \Rightarrow [\Delta A, 0]\in\mathcal{C}_{A,b}^+$. We can force $\Delta b = 0$ by taking the limit as $\theta\to\infty$ in (2.11), giving for the [illegible] minimal backward error in this more limited case

$$\mu_F(y) \equiv \min_{\Delta A\in\mathcal{C}_A^+}\|\Delta A\|_F = \lim_{\theta\to\infty}\mu_F(y,\theta). \tag{2.32}$$

Here we have abused the notation a little by using both $\mu_F(\cdot)$ and $\mu_F(\cdot,\cdot)$. The proof using $\theta\to\infty$ (we use $\varepsilon\equiv\theta^{-2}\searrow 0$) is made possible by a beautiful classical result.

LEMMA 2.3 (Rellich [24, pp. 29–37]; see also Kato [16, pp. 121–122]). [illegible] $\varepsilon\in\mathbb{R}$ [illegible] $H(\varepsilon) = H(\varepsilon)^T\in\mathbb{R}^{n\times n}$ [illegible] $\varepsilon = 0$ [illegible] $\varepsilon = 0$

We need some more results to prepare for Theorem 2.6.

LEMMA 2.4. [illegible] $M = SWS^T$ [illegible] $S\in\mathbb{R}^{m\times n}$, $W = W^T\in\mathbb{R}^{n\times n}$, $m\ge n$, [illegible]

(a) M [illegible] ([illegible]) [illegible] W

(b) [illegible] $W = I - y\alpha y^\dagger$, [illegible] M [illegible]

[illegible]. Part (a) was proven in [15, section 4.5.11] for the case $m = n$. For $m > n$ writing $M = [S,0]\,\mathrm{diag}(W,0)[S,0]^T$ with square $[S,0]$ proves that (a) still holds. Since $I - y\alpha y^\dagger$ has eigenvalues 1 when $y = 0$, and $1-\alpha, 1, \ldots, 1$ when $y\ne 0$, (b) follows from (a). ∎

To make later analysis easier, we use ε to replace θ^{-2}. From (2.12)

$$\rho = \varepsilon/\left(\varepsilon+\|y\|_2^2\right), \qquad \varepsilon+\rho\|y\|_2^2 = \varepsilon(2-\rho), \qquad \varepsilon\equiv\theta^{-2}. \tag{2.33}$$

In our limits we consider only $\varepsilon\ge 0$, so $\varepsilon\to 0$ will always mean $\varepsilon\searrow 0$.

THEOREM 2.5. [illegible] (2.33), [illegible] $A\in\mathbb{R}^{m\times n}$, $0\ne y\in\mathbb{R}^n$, $0\ne b\in\mathbb{R}^m$, $r\equiv b-Ay$,

$$H(\varepsilon)\equiv\varepsilon A(I-yy^\dagger)A^T - r\rho r^T + bb^T,\ \varepsilon\in\mathbb{R},\ H(\varepsilon)w(\varepsilon) = w(\varepsilon)\lambda(\varepsilon),\ w(\varepsilon)\in\mathbb{R}^m, \tag{2.34}$$

$m\ge 2$, [illegible] $\lambda(0)\equiv\lambda_{\min}(H(0))$ [illegible] $\varepsilon\ge 0$ [illegible] $\lambda(\varepsilon)$ [illegible] $w(\varepsilon)$ [illegible]

$$\lambda(\varepsilon) = \lambda_1\varepsilon+\lambda_2\varepsilon^2+\cdots,\ w(\varepsilon) = w_0+w_1\varepsilon+w_2\varepsilon^2+\cdots,\ b^Tw_0 = 0,\ \|w(\varepsilon)\|_2 = 1. \tag{2.35}$$

$$T(\varepsilon)\equiv\varepsilon^{-1}P_b^\perp H(\varepsilon)P_b^\perp, \text{ [illegible] } T(0)\equiv\lim_{\varepsilon\to 0}T(\varepsilon) = P_b^\perp A\left(I-y2y^\dagger\right)A^TP_b^\perp. \tag{2.36}$$

[illegible] $\lambda_*(0)\equiv\lambda_{\min}(T(0))$ [illegible] $\varepsilon\ge 0$ [illegible] $\lambda_*(\varepsilon)$ [illegible] $w_*(\varepsilon)$ [illegible]

$$T(\varepsilon)w_*(\varepsilon) = w_*(\varepsilon)\lambda_*(\varepsilon), \qquad w_*(\varepsilon)\in\mathbb{R}^m, \quad \|w_*(\varepsilon)\|_2 = 1. \tag{2.37}$$

[illegible] $\lim_{\varepsilon\to 0}\varepsilon^{-1}\lambda(\varepsilon)$ [illegible] (2.35)) [illegible] $\lambda_*(0)$ [illegible] $w_*(0)$. (2.37)

$$\lim_{\varepsilon\to 0} b^T w(\varepsilon)=0, \qquad \lim_{\varepsilon\to 0}\varepsilon^{-\frac{1}{2}} b^T w(\varepsilon)=0, \qquad \lim_{\varepsilon\to 0}\varepsilon^{-1} b^T w(\varepsilon)=b^T w_1; \tag{2.38}$$

$$\lim_{\varepsilon\to 0}\varepsilon^{-1}\lambda(\varepsilon)<0 \;\Rightarrow\; \lambda_*(0)<0 \;\Rightarrow\; \lim_{\varepsilon\to 0}\varepsilon^{-1}\lambda(\varepsilon)=\lambda_*(0) \;\;\&\;\; w(0)=\pm w_*(0); \tag{2.39}$$

$$\lim_{\varepsilon\to 0}\varepsilon^{-1}\lambda(\varepsilon)\geq 0 \;\Leftrightarrow\; \lambda_*(0)=0; \tag{2.40}$$

$$\lim_{\varepsilon\to 0}\varepsilon^{-1}\lambda(\varepsilon)=0 \Rightarrow \lambda_*(0)=0 \;\&\; \begin{cases}\{\exists x \text{ [illegible] } b=Ax \;\&\; \angle(x,y)=\pm\pi/4\},\\ \{\varepsilon^{-1}\lambda(\varepsilon)=0 \text{ [illegible] } \varepsilon=0\}.\end{cases} \tag{2.41}$$

[illegible]. The expression for $T(0)$ in (2.36) follows from (2.34) and (2.33), and then

$$T(\varepsilon)=T(0)+P_b^{\perp}Ay\rho y^{\dagger}A^T P_b^{\perp} = P_b^{\perp}A\left[I-y(2-\rho)y^{\dagger}\right]A^T P_b^{\perp}. \tag{2.42}$$

Clearly $H(\varepsilon)$ and $T(\varepsilon)$ are analytic about $\varepsilon=0$, so $w(\varepsilon)$, $\lambda(\varepsilon)$, $w_*(\varepsilon)$, and $\lambda_*(\varepsilon)$ can be chosen to be analytic, with $\|w(\varepsilon)\|_2=\|w_*(\varepsilon)\|_2=1$; see Lemma 2.3. Also $H(\varepsilon)$ can have at most one negative eigenvalue, see the start of the proof of Theorem 2.2, so from Lemma 2.4 $T(\varepsilon)$ can have at most one negative eigenvalue. Since $m\geq 2$, $H(0)=bb^T$ has minimum eigenvalue $\lambda(0)=0$, proving the first part of (2.35). Since $bb^T w(0)=w(0)\lambda(0)=0$, we must have $b^T w(0)=b^T w_0=0$, proving the rest of (2.35). Next (2.35) proves (2.38), and we have

$$\begin{aligned}\lim_{\varepsilon\to 0}\varepsilon^{-1}\lambda(\varepsilon) &= \lim_{\varepsilon\to 0}\varepsilon^{-1}w(\varepsilon)^T H(\varepsilon)w(\varepsilon) = \lim_{\varepsilon\to 0}\varepsilon^{-\frac{1}{2}}w(\varepsilon)^T P_b^{\perp}H(\varepsilon)P_b^{\perp}w(\varepsilon)\varepsilon^{-\frac{1}{2}}\\ &= \lim_{\varepsilon\to 0} w(\varepsilon)^T T(\varepsilon)w(\varepsilon) = w(0)^T T(0)w(0)\geq \lambda_*(0)=\lambda_{\min}(T(0)),\end{aligned} \tag{2.43}$$

so $\lim_{\varepsilon\to 0}\varepsilon^{-1}\lambda(\varepsilon)<0 \Rightarrow \lambda_*(0)<0$. When $\lambda_*(0)<0$, it is a singleton; see Lemma 2.4, so for small enough ε, $\lambda_*(\varepsilon)<0$; then $b^T w_*(\varepsilon)=0$ from (2.37) and (2.36), giving

$$\lambda_*(\varepsilon)=\varepsilon^{-1}w_*(\varepsilon)^T P_b^{\perp}H(\varepsilon)P_b^{\perp}w_*(\varepsilon)=\varepsilon^{-1}w_*(\varepsilon)^T H(\varepsilon)w_*(\varepsilon). \tag{2.44}$$

Taking the limit as $\varepsilon\to 0$ and using (2.43), with $\|w(\varepsilon)\|_2=\|w_*(\varepsilon)\|_2=1$ gives

$$\begin{aligned}\lambda_*(0) &= \lim_{\varepsilon\to 0}\varepsilon^{-1}w_*(\varepsilon)^T H(\varepsilon)w_*(\varepsilon)\geq \lim_{\varepsilon\to 0}\varepsilon^{-1}\lambda(\varepsilon)=w(0)^T T(0)w(0)\\ &\geq \lambda_*(0)=w_*(0)^T T(0)w_*(0),\end{aligned}$$

proving equality throughout, so that $\lim_{\varepsilon\to 0}\varepsilon^{-1}\lambda(\varepsilon)=\lambda_*(0)$ when $\lambda_*(0)<0$. Also $w(0)^T T(0)w(0)=w_*(0)^T T(0)w_*(0)$ is a minimum of $w^T T(0)w$ over $w^T w=1$ with unique minimizer (up to sign) when $\lambda_*(0)$ is a singleton, completing the proof of (2.39). Since $T(\varepsilon)b=0$, we see that $\lambda_*(0)\leq 0$, and (2.40) follows using (2.39).

Now assume that $\lim_{\varepsilon\to 0}\varepsilon^{-1}\lambda(\varepsilon)=0$, ($\lambda_1=0$ in (2.35)). Then $\lambda_*(0)=0$ in (2.41) follows from (2.40). If $[A,b]$ has rank s, then $\varepsilon^{-1}H(\varepsilon)$ has m–s trivial zero eigenvalues. If in the limit as $\varepsilon\to 0$ there are only trivial zero eigenvalues, then by continuity, $\varepsilon^{-1}\lambda(\varepsilon)=0$ in some neighborhood of $\varepsilon=0$. Next assume that there is a nontrivial zero eigenvalue, that is, an eigenpair of the form (2.35), with $\lambda_1=0$ and $A^T w_0\neq 0$. But $\lambda_*(0)=0$ shows that $T(0)$ is positive semidefinite, and $0=\lim_{\varepsilon\to 0}\varepsilon^{-1}\lambda(\varepsilon)=$

$\lim_{\varepsilon\to 0} w_0^T\varepsilon^{-1}H(\varepsilon)w_0 = \lim_{\varepsilon\to 0} w_0^T T(\varepsilon)w_0$; see (2.36), so $0 = w_0^T T(0)w_0 = w_0^T A(I - y2y^\dagger)A^T w_0$. Here $I - y2y^\dagger$ is a Householder reflection. Thus

$$\left\|A^T w_0\right\|_2^2 = 2\left(w_0^T Ay\right)^2/\|y\|_2^2, \qquad 0 = T(0)w_0 = P_b^\perp A\left(I - y2y^\dagger\right)A^T w_0, \tag{2.45}$$

and $A(I-y2y^\dagger)A^T w_0 = bb^\dagger A(I-y2y^\dagger)A^T w_0 \neq 0$; see (2.2). Then $Ax\nu = b\nu$, where

$$x\nu = \left(I - y2y^\dagger\right)A^T w_0, \quad \nu \equiv b^T A\left(I - y2y^\dagger\right)A^T w_0/b^T b \neq 0.$$

From this $y^T x\nu = -y^T A^T w_0$, and with (2.45) $x^T x\nu^2 = \|A^T w_0\|_2^2 = 2(w_0^T Ay)^2/\|y\|_2^2 = 2(y^T x)^2\nu^2/\|y\|_2^2$. This gives $2(y^T x)^2 = y^T y \cdot x^T x$, so $\angle(x,y) = \pm\pi/4$, proving (2.41). □

2.2. In (2.41) the case $b = Ax$ & $\angle(x,y) = \pm\pi/4$ is extremely unlikely (it has "probability zero"), requiring $b \in \mathcal{R}(A)$ (a highly unlikely situation when we are solving DLS problems) and y to be a terrible approximation to the unique x for which $b = Ax$, giving $\angle(x,y) = \pm\pi/4$.

We can now obtain the equivalent of Theorem 2.2 for a backward perturbation restricted to A alone.

THEOREM 2.6. $A \in \mathbb{R}^{m\times n}$, $m \geq 2$, $b \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, (2.2) $r \equiv b - Ay$

$$N_\infty \equiv \left[\left(I-bb^\dagger\right)A\left(I-yy^\dagger\right), \frac{\|r\|_2}{\|y\|_2}\left(I-bb^\dagger\right), \frac{b}{\|b\|_2}\frac{\|r\|_2}{\|y\|_2}\right], \tag{2.46}$$

$$M_\infty = M_\infty(y) \equiv \left(I-bb^\dagger\right)A\left(I-y2y^\dagger\right)A^T\left(I-bb^\dagger\right) = N_\infty N_\infty^T - \|r\|_2^2/\|y\|_2^2\, I. \tag{2.47}$$

$\lambda_{\min}(M_\infty) \leq 0$ M_∞ $\mu_F(y)$ (2.32)

$$\mu_F^2(y) = \frac{\|r\|_2^2}{\|y\|_2^2} + \lambda_{\min}(M_\infty) = \sigma_{\min}^2(N_\infty). \tag{2.48}$$

$\mu_F(y)$

$$\widehat{\Delta A} = \begin{cases} ry^\dagger & \lambda_{\min}(M_\infty) = 0, \\ ry^\dagger - w_* w_*^\dagger A(I - y2y^\dagger) & \lambda_{\min}(M_\infty) < 0, \end{cases} \tag{2.49}$$

w_* M_∞ $\lambda_{\min}(M_\infty) < 0$ N_∞ $\sigma_{\min}(N_\infty)$ $\lambda_{\min}(M_\infty) < 0$. (2.47) $\lambda_{\min}(M_\infty) = \lim_{\theta\to\infty}\lambda_{\min}(M(\theta))$. (2.14) (2.48)–(2.49) (2.15)–(2.16) $\theta\to\infty$ 2.2 $\lim_{\theta\to\infty}\lambda_{\min}(M(\theta)) = 0$. 2.1

. Since (2.47) follows from (2.46), the results on N_∞ follow trivially from those on M_∞. Since $M_\infty b = 0$, $\lambda_{\min}(M_\infty) \leq 0$. From Lemma 2.4 M_∞ has at most one negative eigenvalue. In fact M_∞ in (2.47) is identical to $T(0)$ in (2.36), so with $w_* \equiv w_*(0)$, $\lambda_* \equiv \lambda_{\min}(M_\infty) \equiv \lambda_*(0)$ in (2.37), $M_\infty w_* = w_*\lambda_*$. From (2.14) and (2.34)

$$M(\theta) \equiv A\left(I-yy^\dagger\right)A^T - r\rho\theta^2 r^T + b\,\theta^2 b^T = \varepsilon^{-1}H(\varepsilon), \quad \text{with } \varepsilon \equiv \theta^{-2}. \tag{2.50}$$

Since $M(\theta)$ can have at most one negative eigenvalue, when $\lim_{\theta\to\infty}\lambda_{\min}(M(\theta)) < 0$, $\lambda_{\min}(M(\theta))$ is the unique minimum eigenvalue for large enough θ, and w_θ in (2.16)

can be taken as $w(\varepsilon)$ in Theorem 2.5. This, and noting that $\lambda(\theta^2 H(\theta^{-2}))$ is equal to $\lambda_{\min}(M(\theta))$ for large enough θ, gives from (2.38)–(2.41)

$$\lim_{\theta\to\infty}\lambda_{\min}(M(\theta))<0 \;\Rightarrow\; \begin{cases} \lambda_{\min}(M_\infty)=\lim_{\theta\to\infty}\lambda_{\min}(M(\theta)), \\ \;\&\;\; w_*=\pm\lim_{\theta\to\infty}w_\theta \;\;\&\;\; \lim_{\theta\to\infty}\theta w_\theta^T b=0; \end{cases}$$

$$\lim_{\theta\to\infty}\lambda_{\min}(M(\theta))\geq 0 \;\Leftrightarrow\; \lambda_*\equiv\lambda_{\min}(M_\infty)=0;$$

$$\lim_{\theta\to\infty}\lambda_{\min}(M(\theta))=0 \;\Rightarrow\; \lambda_{\min}(M_\infty)=0 \;\&\; \begin{cases} \{\exists x \;:\; b=Ax \;\&\; \angle(x,y)=\pm\pi/4\}, \\ \text{or } \{\exists\theta_1 \;:\; \lambda_{\min}(M(\theta))=0 \;\forall\theta>\theta_1\}. \end{cases}$$

But from (2.12) $\lim_{\theta\to\infty}\rho=0$ and $\lim_{\theta\to\infty}\rho\theta^2=\|y\|_2^{-2}$, so that (2.48) is the limiting value of both cases of (2.15) as $\theta\to\infty$. If $\lim_{\theta\to\infty}\lambda_{\min}(M(\theta))\neq 0$, it can also be seen that in the limit the two criteria in (2.16) become the respective criteria in (2.49), where if $\lim_{\theta\to\infty}\lambda_{\min}(M(\theta))<0$, $b^T w_*=\lim_{\theta\to\infty}b^T w_\theta=0$, so that in the limit the two expressions for $\widehat{\Delta A}$ in (2.16) become the respective expressions in (2.49). If $\lim_{\theta\to\infty}\lambda_{\min}(M(\theta))=0$ and this corresponds to trivial zero eigenvalues only, the top row of (2.49) is clearly once again the correct limit. Only the probability zero case of $\lim_{\theta\to\infty}\lambda_{\min}(M(\theta))=0$ with $b=Ax$, $\angle(x,y)=\pm\pi/4$, allows the possibility that $\lambda_{\min}(M(\theta))<0$ for arbitrarily large θ, and since in this case $\lambda_{\min}(M_\infty)=0$, this suggests that (2.16) could fail to give the correct limiting perturbation in (2.49). That is, although (2.49) is correct, until proven otherwise there remains the possibility that taking the limit in (2.16) could lead to $\widehat{\Delta A}=ry^\dagger-w_*w_*^\dagger A(I-y2y^\dagger)$ rather than $ry^\dagger$ in this one strange case; see (2.49). □

Deriving $\mu_F(y)$ directly as we did for $\mu_F(y,\theta)$ also leads to the results down to the sentence including (2.49). But Theorem 2.5 describes the limiting behavior as well.

To parallel the remark given in [13, section 20.7] for some formulas for the minimal backward error of ordinary least squares problems, computing and adding the eigenvalue in (2.15) or (2.48) is not wise computationally. Catastrophic cancellation may occur when it is negative. Furthermore, the computed value may have very poor accuracy even using well-known software such as MATLAB 7.4, e.g., in (2.48) the computed value of $\lambda_{\min}(M_\infty)$ may be smaller than $-\|r\|_2^2/\|y\|_2^2$. The singular value is much more reliable for computation. If we computed that using the Golub–Reinsch singular value decomposition algorithm, it would need about $8/3m^3+4mn^2$ flops, but one point of this paper is that we can use cheaper lower bounds or estimates instead.

In Theorem 2.2 we have either $\lambda_{\min}(M(\theta))<0$ or $\lambda_{\min}(M(\theta))\geq 0$, while in Theorem 2.6 we have either $\lambda_{\min}(M_\infty)<0$ or $\lambda_{\min}(M_\infty)=0$. By substituting the resulting perturbations in the relevant inequalities, it is straightforward to see that the inequality in (2.9) is satisfied when $\lambda_{\min}(M(\theta))\geq 0$ and $\operatorname{rank}(A+r(1-\rho)y^\dagger)=n$, and the inequality in (2.31) is satisfied when $\lambda_{\min}(M_\infty)=0$ and $\operatorname{rank}(A+ry^\dagger)=n$. It follows that, in these two special cases, the extended minimal backward error is actually the true minimal backward error and that nothing was lost by using the "supersets" $\mathcal{C}_{A,b}^+$ and $\mathcal{C}_A^+$. We will supply further justification for the use of these supersets later.

The following result indicates that the extended minimal backward error $\mu_F(y)$ is continuous at $y=\hat{x}$, where $\hat{x}$ is the DLS solution in (2.5)–(2.6), where of course $\mu_F(\hat{x})=0$. In order to save space, here and in the rest of the paper we will only consider the case where A is perturbed, but all of the results could be extended to the more general case where both A and b are perturbed.

COROLLARY 2.7. [illegible] 2.6 [illegible] $\hat{x}$ [illegible] (2.5)–(2.6) [illegible] $\hat{M}_\infty\equiv M_\infty(\hat{x})$ ([illegible] (2.47)) [illegible] $\hat{r}\equiv b-A\hat{x}$ [illegible]

... (2.15)

$$\lim_{y\to\hat{x}} \mu_F(y) = \mu_F(\hat{x}) = \left(\frac{\|\hat{r}\|_2^2}{\|\hat{x}\|_2^2} + \lambda_{\min}\left(\hat{M}_\infty\right)\right)^{1/2} = 0.$$

... First we see that (2.5) just says

$$A^T\hat{r} = A^T(b - A\hat{x}) = -\hat{x}\|b - A\hat{x}\|_2^2/\|\hat{x}\|_2^2 = -\hat{x}\|\hat{r}\|_2^2/\|\hat{x}\|_2^2,$$

and multiplying this on the left by $\hat{x}^T$ shows that

$$0 = (b - A\hat{x})^T(b - A\hat{x}) + (A\hat{x})^T(b - A\hat{x}) = b^T(b - A\hat{x}) = b^T\hat{r} \tag{2.51}$$

so that $(I - bb^\dagger)\hat{r} = \hat{r}$. Since $\hat{M}_\infty = M_\infty(\hat{x}) = (I - bb^\dagger)A(I - 2\hat{x}\hat{x}^\dagger)A^T(I - bb^\dagger)$, the above give

$$\begin{aligned}\hat{M}_\infty\hat{r} &= \left(I - bb^\dagger\right) A \left(I - 2\hat{x}\hat{x}^\dagger\right) A^T\hat{r} = -\left(I - bb^\dagger\right) A \left(I - 2\hat{x}\hat{x}^\dagger\right)\hat{x}\|\hat{r}\|_2^2/\|\hat{x}\|_2^2 \\ &= \left(I - bb^\dagger\right) A\hat{x}\|\hat{r}\|_2^2/\|\hat{x}\|_2^2 = \left(I - bb^\dagger\right)(A\hat{x} - b)\|\hat{r}\|_2^2/\|\hat{x}\|_2^2 \\ &= -\hat{r}\|\hat{r}\|_2^2/\|\hat{x}\|_2^2.\end{aligned}$$

Thus by Lemma 2.4, $-\|\hat{r}\|_2^2/\|\hat{x}\|_2^2$ is the only negative eigenvalue of $\hat{M}_\infty$, and $\mu_F(\hat{x}) = 0$. Clearly when $y \to \hat{x}$, we have $r = b - Ay \to \hat{r}$, $M_\infty \to \hat{M}_\infty$, and by the continuity of the eigenvalues of M_∞ in (2.47), $\lambda_{\min}(M_\infty) \to \lambda_{\min}(\hat{M}_\infty)$, completing the proof. □

Since in (2.32) $\mathcal{C}_A \subseteq \mathcal{C}_A^+$,

$$\mu_F(y) \equiv \min_{\Delta A\in\mathcal{C}_A^+} \|\Delta A\|_F \le \min_{\Delta A\in\mathcal{C}_A} \|\Delta A\|_F,$$

i.e., $\mu_F(y)$ is a lower bound on the minimal backward error. However we have found computationally, see section 5, that when y is a reasonable approximation to the exact solution of the DLS problem (2.1), the minimal perturbation $\widehat{\Delta A}$ usually satisfies the inequality in (2.31). Therefore in such cases $\mu(y)$ is actually the minimal backward error. The following result partially justifies this finding.

THEOREM 2.8. ... $A \in \mathbb{R}^{m\times n}$ ... $b \in \mathbb{R}^m$ ... (2.2) ... $\hat{x}$ ... (2.1) ... $\epsilon > 0$ ... $\|y - \hat{x}\|_2 < \epsilon$, ... $\mu_F(y)$ ...

... For any given y, Theorem 2.6 shows that $\widehat{\Delta A}$ satisfying (2.49) is the minimizer of (2.32). Notice that when $y \to \hat{x}$, we have from Corollary 2.7 that $\widehat{\Delta A} \to 0$. Thus

$$\lim_{y\to\hat{x}} \left(\frac{\left\|b - \left(A + \widehat{\Delta A}\right) y\right\|_2}{\|y\|_2} - \sigma_{\min}\left(A + \widehat{\Delta A}\right)\right) = \frac{\|b - A\hat{x}\|_2}{\|\hat{x}\|_2} - \sigma_{\min}(A).$$

Since $\frac{\|b-A\hat{x}\|_2}{\|\hat{x}\|_2} - \sigma_{\min}(A) < 0$, there must exist $\epsilon > 0$ such that when $\|y - \hat{x}\|_2 < \epsilon$,

$$\frac{\left\|b - \left(A + \widehat{\Delta A}\right) y\right\|_2}{\|y\|_2} - \sigma_{\min}\left(A + \widehat{\Delta A}\right) < 0.$$

Therefore $\widehat{\Delta A} \in \mathcal{C}_A$ and $\mu_F(y) = \min_{\Delta A\in\mathcal{C}_A} \|\Delta A\|_F$, i.e., when $\|y - \hat{x}\|_2 < \epsilon$, $\mu_F(y)$ is the true minimal backward error. □

3. A lower bound on $\min_{\Delta A\in\mathcal{C}_A^+}\|\Delta A\|_2$. Since computing $\mu_F(y)$ directly is expensive, in this section we suggest a good lower bound which can be estimated easily.

First we give the following result, which is analogous to Theorem 3.1 in [35] for ordinary least squares problems.

THEOREM 3.1. [illegible] 2.6 [illegible] $\mathcal{C}_A^+$ [illegible] (2.30) [illegible] $\mu_2(y)\equiv\min_{\Delta A\in\mathcal{C}_A^+}\|\Delta A\|_2$ [illegible] $\mu_F(y)$ [illegible] (2.32),

$$\frac{1}{\sqrt{2}}\mu_F(y)\le\mu_2(y)\le\mu_F(y). \tag{3.1}$$

[illegible] For any $\Delta A\in\mathcal{C}_A^+$, we see from Lemma 2.1 that there exist w satisfying $b^Tw=0$ and $Z\in\mathbb{R}^{m\times n}$ such that

$$\begin{aligned}\Delta A&=(b-w)y^\dagger+\left(I-ww^\dagger\right)Z\left(I-yy^\dagger\right)-A\\&=\left[\left(I-ww^\dagger\right)b-w\right]y^\dagger+\left(I-ww^\dagger\right)Z\left(I-yy^\dagger\right)\\&\quad-\left(I-ww^\dagger\right)Ayy^\dagger-ww^\dagger Ayy^\dagger-\left(I-ww^\dagger\right)A\left(I-yy^\dagger\right)-ww^\dagger A\left(I-yy^\dagger\right)\\&=\left[\left(I-ww^\dagger\right)(b-Ay)-w-ww^\dagger Ay\right]y^\dagger-ww^\dagger A\left(I-yy^\dagger\right)\\&\quad+\left(I-ww^\dagger\right)(Z-A)\left(I-yy^\dagger\right).\end{aligned}$$

Denote $\Delta A_1=[(I-ww^\dagger)(b-Ay)-w-ww^\dagger Ay]y^\dagger$, $\Delta A_2=-ww^\dagger A(I-yy^\dagger)$, and $\Delta A_3=(I-ww^\dagger)(Z-A)(I-yy^\dagger)$. Since $\Delta A_1\Delta A_2^T=0$, $\Delta A_1\Delta A_3^T=0$ and $\Delta A_2^T\Delta A_3=0$ and ΔA_1 and ΔA_2 are rank 1 matrices,

$$\begin{aligned}\|\Delta A\|_2^2&=\|\Delta A_1+\Delta A_2+\Delta A_3\|_2^2=\|(\Delta A_1+\Delta A_2+\Delta A_3)(\Delta A_1+\Delta A_2+\Delta A_3)^T\|_2\\&=\left\|\Delta A_1\Delta A_1^T+(\Delta A_2+\Delta A_3)(\Delta A_2+\Delta A_3)^T\right\|_2\\&\ge\max\left\{\left\|\Delta A_1\Delta A_1^T\right\|_2,\|(\Delta A_2+\Delta A_3)(\Delta A_2+\Delta A_3)^T\|_2\right\}\\&\ge\frac{1}{2}\left(\|\Delta A_1\|_2^2+\|\Delta A_2+\Delta A_3\|_2^2\right)\\&=\frac{1}{2}\left(\|\Delta A_1\|_2^2+\|(\Delta A_2+\Delta A_3)^T(\Delta A_2+\Delta A_3)\|_2\right)\\&=\frac{1}{2}\left(\|\Delta A_1\|_2^2+\left\|\Delta A_2^T\Delta A_2+\Delta A_3^T\Delta A_3\right\|_2\right)\ge\frac{1}{2}\left(\|\Delta A_1\|_2^2+\|\Delta A_2\|_2^2\right)\\&=\frac{1}{2}\left(\|\Delta A_1\|_F^2+\|\Delta A_2\|_F^2\right)=\frac{1}{2}\|\Delta A_1+\Delta A_2\|_F^2\ge\frac{1}{2}\min_{\Delta A\in\mathcal{C}_A^+}\|\Delta A\|_F^2,\end{aligned}$$

where the last inequality is due to the fact that $\Delta A_1+\Delta A_2\in\mathcal{C}_A^+$ (take $Z=A$). Therefore

$$\min_{\Delta A\in\mathcal{C}_A^+}\|\Delta A\|_2^2\ge\frac{1}{2}\min_{\Delta A\in\mathcal{C}_A^+}\|\Delta A\|_F^2,$$

leading to the first inequality in (3.1). The second inequality in (3.1) is easy to prove. In fact, if $\widehat{\Delta A}$ is a minimal solution to (2.32), then

$$\mu_F(y)=\left\|\widehat{\Delta A}\right\|_F\ge\left\|\widehat{\Delta A}\right\|_2\ge\mu_2(y).\qquad\square$$

Now we give a lower bound on $\mu_2(y)\equiv\min_{\Delta A\in\mathcal{C}_A^+}\|\Delta A\|_2$.

THEOREM 3.2. [illegible] 2.6 [illegible] $\mu_2(y) \equiv \min_{\Delta A \in \mathcal{C}_A^+} \|\Delta A\|_2$ [illegible] $\mathcal{C}_A^+$ [illegible] (2.30)

$$\mu_2(y) \geq \mu_2^{\mathrm{lb}}(y) \equiv \frac{2\beta_0}{\beta_1 + \sqrt{\beta_1^2 + 4\beta_0}}, \tag{3.2}$$

[illegible]

$$\beta_0 \equiv \frac{\left\|(A^T r\|y\|_2^2 + y\|r\|_2^2)\right\|_2}{2\|y\|_2^3}, \quad \beta_1 \equiv \frac{\|y\|_2^3\|A\|_2 + 3\|y\|_2^2\|r\|_2}{2\|y\|_2^3}. \tag{3.3}$$

[illegible]. For any $\Delta A \in \mathcal{C}_A^+$, from (2.30) we obtain

$$(A + \Delta A)^T (r - \Delta A y)\|y\|_2^2 + y\|r - \Delta A y\|_2^2 = 0.$$

Thus we have

$$\begin{aligned} A^T r\|y\|_2^2 + y\|r\|_2^2 = A^T \Delta A y\|y\|_2^2 - \Delta A^T r\|y\|_2^2 + y2\left(r^T \Delta A y\right) \\ + \Delta A^T \Delta A y\|y\|_2^2 - y\|\Delta A y\|_2^2. \end{aligned}$$

Then taking the 2-norm on both sides of this equation, we obtain the inequality

$$\left\|\left(A^T r\|y\|_2^2 + y\|r\|_2^2\right)\right\|_2 \leq \left(\|y\|_2^3\|A\|_2 + 3\|y\|_2^2\|r\|_2\right)\|\Delta A\|_2 + 2\|y\|_2^3\|\Delta A\|_2^2,$$

that is, with (3.3), the quadratic inequality in terms of $\xi \equiv \|\Delta A\|_2$:

$$\beta_0 \leq \beta_1 \xi + \xi^2.$$

Since ξ and β_1 are nonnegative, $\xi \geq \xi_+$, where ξ_+ is the positive root of $\beta_0 = \beta_1 \xi + \xi^2$, so

$$\xi \geq \xi_+ = \left(\sqrt{\beta_1^2 + 4\beta_0} - \beta_1\right)/2 = 2\beta_0 / \left(\sqrt{\beta_1^2 + 4\beta_0} + \beta_1\right),$$

giving (3.2). ∎

The lower bound in (3.2) can usually be evaluated in $\mathcal{O}(mn)$ flops, since $\|A\|_2$ can usually be estimated by a standard norm estimator in $O(mn)$ flops; see [13, section 15.2]. In fact a good estimate of $\|A\|_2$ might already be available from whatever method is used for obtaining y, and the cost will essentially be the $4mn$ flops for computing $A^T(b - Ay)$.

Also $\mu_2^{\mathrm{lb}}(\hat{x}) = \mu_2(\hat{x}) = \mu_F(\hat{x}) = 0$ as desired; see (2.5), Corollary 2.7, and (3.1).

4. An asymptotic estimate for $\mu_F(y)$. Computing $\mu_F(y)$ directly is expensive, and the lower bound (3.2) may not be very tight. In this section we would like to give an asymptotic estimate by following the general approach given in [9].

Let $f(A, y) \equiv (b - Ay)^T(b - Ay)y + (y^T y)A^T(b - Ay) = \|r\|_2^2 y + \|y\|_2^2 A^T r$. Note that $f(A, \hat{x}) = 0$ (see (2.5)). The extended minimal backward perturbation ΔA is the matrix satisfying $f(A + \Delta A, y) = 0$ and $\mu_F(y) = \|\Delta A\|_F$. But by Taylor's expansion, for small enough $E \in \mathbb{R}^{m \times n}$,

$$f(A + E, y) \approx f(A, y) + \mathcal{J}_A f(A, y)\mathtt{vec}(E),$$

where $\mathcal{J}_A f(A,y) \in \mathbb{R}^{n\times mn}$ is the Jacobian matrix of f with respect to $\mathtt{vec}(A)$. Thus an approximation to ΔA is that E giving the minimum 2-norm solution to

$$f(A,y) + \mathcal{J}_A f(A,y)\mathtt{vec}(E) = 0, \tag{4.1}$$

that is, E such that

$$\mathtt{vec}(E) = -[\mathcal{J}_A f(A,y)]^\dagger f(A,y), \quad \tilde{\mu}_F(y) \equiv \|E\|_F = \|[\mathcal{J}_A f(A,y)]^\dagger f(A,y)\|_2. \tag{4.2}$$

THEOREM 4.1. [illegible] 2.6, $\tilde{\mu}_F(y)$ [illegible] (4.2) [illegible] $\mu_F(y)$ [illegible] (2.32) [illegible] $\hat{x}$ [illegible] (2.1),

$$\lim_{y\to\hat{x}} \frac{\tilde{\mu}_F(y)}{\mu_F(y)} = 1.$$

[illegible]. By Taylor's expansion,

$$0 = f(A+\Delta A, y) = f(A,y) + \mathcal{J}_A f(A,y)\mathtt{vec}(\Delta A) + O\left(\|\Delta A\|_F^2\right). \tag{4.3}$$

Thus from (4.2),

$$\mathtt{vec}(E) = -[\mathcal{J}_A f(A,y)]^\dagger f(A,y) = [\mathcal{J}_A f(A,y)]^\dagger \mathcal{J}_A f(A,y)\mathtt{vec}(\Delta A) + O\left(\|\Delta A\|_F^2\right).$$

Taking the 2-norm and noticing $[\mathcal{J}_A f(A,y)]^\dagger \mathcal{J}_A f(A,y)$ is an orthogonal projection matrix, we obtain

$$\tilde{\mu}_F(y) \le \mu_F(y) + O\left(\|\Delta A\|_F^2\right),$$

which, with Corollary 2.7, leads to $\lim_{y\to\hat{x}} \tilde{\mu}_F(y)/\mu_F(y) \le 1$.

On the other hand, from (4.1) and (4.3) we can obtain

$$\mathcal{J}_A f(A,y)\mathtt{vec}(\Delta A) = \mathcal{J}_A f(A,y)\left[\mathtt{vec}(E) + O\left(\|\Delta A\|_F^2\right)\right].$$

Since ΔA is a matrix satisfying the above equality with minimum F-norm, we must have

$$\|\mathtt{vec}(\Delta A)\|_2 \le \left\|\mathtt{vec}(E) + O\left(\|\Delta A\|_F^2\right)\right\|_2 \le \|\mathtt{vec}(E)\|_2 + O\left(\|\Delta A\|_F^2\right),$$

which, with Corollary 2.7, leads to $\lim_{y\to\hat{x}} \tilde{\mu}_F(y)/\mu_F(y) \ge 1$, completing the proof. □

Theorem 4.1 is similar to [9, Corollary 3.4], where a general minimal backward error problem was considered, and applying the corollary to our case will result in the asymptotic estimate $\|[\mathcal{J}_A f(A,\hat{x})]^\dagger f(A,y)\|_2$. We thank the referee who pointed out that a general version of Theorem 4.1 was given in [10] (with no formal proof).

In the following we will consider computing or estimating $\tilde{\mu}_F(y)$. First we would like to obtain an explicit expression for it. If $f = (f_i)$ and $g = (g_i)$ are column vectors, then we define the matrix $\partial f/\partial g^T \equiv (\partial f_i/\partial g_j)$. Write $m \times n$ $A = [a_1, \ldots, a_n]$, $y^T = (\eta_1, \ldots, \eta_n)$, then $\partial r/\partial a_j^T = \partial(b - Ay)/\partial a_j^T = -\eta_j I$, $\partial(r^T r)/\partial a_j^T = 2r^T \partial r/\partial a_j^T = -2\eta_j r^T$ and if $i \ne j$, $\partial(a_i^T r)/\partial a_j^T = -\eta_j a_i^T$, while $\partial(a_j^T r)/\partial a_j^T = r^T - \eta_j a_j^T$, from

which we see that

$$\begin{aligned}
&\mathcal{J}_A f(A,y) \equiv \partial f(A,y)/\partial \mathtt{vec}(A)^T = \left[\partial f(A,y)/\partial a_1^T, \ldots, \partial f(A,y)/\partial a_n^T\right],\\
&\partial f(A,y)/\partial a_j^T = \partial\left(r^T r y + y^T y A^T r\right)/\partial a_j^T = -2\eta_j y r^T + y^T y e_j r^T - \eta_j y^T y A^T,\\
&\mathcal{J}_A f(A,y)\cdot[\mathcal{J}_A f(A,y)]^T = \sum_{j=1}^{n}\left[\partial f(A,y)/\partial a_j^T\right]\cdot\left[\partial f(A,y)/\partial a_j^T\right]^T\\
&= \sum_{j=1}^{n}\left(-2\eta_j y r^T + y^T y e_j r^T - \eta_j y^T y A^T\right)\left(-2\eta_j y r^T + y^T y e_j r^T - \eta_j y^T y A^T\right)^T\\
&= \|y\|_2^6\left[A^T A + A^T r y^\dagger + \left(y^\dagger\right)^T r^T A + \left(\|r\|_2^2/\|y\|_2^2\right) I\right]\\
&= \|y\|_2^6\begin{bmatrix} A + r y^\dagger \\ \left(\|r\|_2/\|y\|_2\right)\left(I - y y^\dagger\right)\end{bmatrix}^T\begin{bmatrix} A + r y^\dagger \\ \left(\|r\|_2/\|y\|_2\right)\left(I - y y^\dagger\right)\end{bmatrix}. \qquad (4.4)
\end{aligned}$$

Here the matrix $\begin{bmatrix} A+ry^\dagger \\ (\|r\|_2/\|y\|_2)(I-yy^\dagger)\end{bmatrix}$ has full column rank. In fact, if it does not, then there exists nonzero $x \in \mathbb{R}^n$ such that

$$\left(A + r y^\dagger\right) x = 0, \quad \left(I - y y^\dagger\right) x = 0,$$

and it follows that $Ay + r = 0$, so $b = 0$, contradicting our assumption (2.2). Therefore $\mathcal{J}_A f(A,y)$ has full row rank. Then from (4.2),

$$\begin{aligned}
\tilde{\mu}_F(y) &= \left\|[\mathcal{J}_A f(A,y)]^T\left\{\mathcal{J}_A f(A,y)\cdot[\mathcal{J}_A f(A,y)]^T\right\}^{-1} f(A,y)\right\|_2\\
&= \left\|\left\{\mathcal{J}_A f(A,y)\cdot[\mathcal{J}_A f(A,y)]^T\right\}^{-1/2} f(A,y)\right\|_2, \qquad (4.5)
\end{aligned}$$

where the second equality can easily be proved by using the SVD of $\mathcal{J}_A f(A,y)$.

Define

$$B \equiv \begin{bmatrix} A + r y^\dagger \\ \left(\|r\|_2/\|y\|_2\right)\left(I - y y^\dagger\right)\end{bmatrix}, \quad c \equiv \begin{bmatrix} r \\ 0\end{bmatrix} \in \mathbb{R}^{m+n}. \qquad (4.6)$$

Note that

$$f(A,y) = \|y\|_2^2\left(A^T r + \|r\|_2^2\left(y^\dagger\right)^T\right) = \|y\|_2^2 B^T c. \qquad (4.7)$$

Then from (4.5) with (4.4) and (4.7), it follows that

(4.8)

$$\tilde{\mu}_F(y) = \frac{\|\left(B^T B\right)^{-1/2} B^T c\|_2}{\|y\|_2} = \frac{[c^T B\left(B^T B\right)^{-1} B^T c]^{1/2}}{\|y\|_2} = \frac{\|B\left(B^T B\right)^{-1} B^T c\|_2}{\|y\|_2}.$$

Note that $B(B^TB)^{-1}B^T$ is an orthogonal projector onto $\mathcal{R}(B)$.

The asymptotic estimate $\tilde{\mu}_F(y)$ is analogous to an estimate for the minimal backward error for ordinary least squares problems whose various forms have been studied in [9], [11], [12], and [17]. One method for computing $\tilde{\mu}_F(y)$ is to use the QR factorization. If $B = QR$ where $Q \in \mathbb{R}^{(m+n)\times n}$ satisfies $Q^TQ = I_n$ and R is upper triangular, then we see that

$$\tilde{\mu}_F(y) = \left\|Q^T c\right\|_2 / \|y\|_2. \qquad (4.9)$$

If we use Householder QR factorization, this method will cost $2(m + 2/3n^2)n^2$ flops. The other method is to use the moment method (see, e.g., [6]) by following [27, Part I]. For brevity, we will not give details here.

5. Numerical tests. In section 2 we gave an extended minimal backward error $\mu_F(y)$, which is a lower bound on the minimal backward error. But if the inequality in (2.31) holds for the extended minimal backward perturbation $\widehat{\Delta A}$ given in (2.49), then $\mu_F(y)$ is in fact the minimal backward error. Our numerical tests indicate that if the given vector y is a reasonable approximation to the true DLS solution, the inequality in (2.31) holds, where ΔA is the minimal $\widehat{\Delta A}$ given in (2.16). We will give some examples in this section to illustrate this. In section 3 we gave a lower bound $\mu_2^{\mathrm{lb}}(y)$ on $\mu_2(y)$, which is also a lower bound on $\mu_F(y)$ (since $\mu_2(y) \le \mu_F(y)$). In section 4 we presented an asymptotic estimate $\tilde{\mu}_F(y)$ of $\mu_F(y)$. We will give numerical examples to show how good $\mu_2^{\mathrm{lb}}(y)$ and $\tilde{\mu}_F(y)$ are as approximations to $\mu_F(y)$. We carried out computations using MATLAB 7.4 on a MacBook running Mac OS X 10.4.11.

In our numerical tests the data was constructed as follows (`randn` and `rand` are two MATLAB built-in functions for generating random matrices with normal and uniform distributions, respectively):

- We use two types of test matrix for A:
  Type 1: $A = \tilde{A}/\|\tilde{A}\|_F$, $\tilde{A} = \mathtt{randn}(100, 40)$. Typically $\kappa_2(A) \le 10$.
  Type 2: $A = \tilde{A}/\|\tilde{A}\|_F$, $\tilde{A} = U\Sigma V^T$, 40×40 $\Sigma = \mathrm{diag}(\sigma_i)$, $\sigma_i = 10^{-4(i-1)/39}$, $U \in \mathbb{R}^{100\times 40}$ and $V \in \mathbb{R}^{40\times 40}$ are the Q-factors of the QR factorizations of two random matrices $\mathtt{randn}(100, 40)$ and $\mathtt{randn}(40, 40)$, respectively. Note that $\kappa_2(A) = 10^4$.
- $b = (A + E)x$, $x = [1, \ldots, 1]^T \in \mathbb{R}^{40}$, $E = \frac{\delta_A}{\sqrt{100\times 40}}\mathtt{rand}(100, 40)$ (note that $\|E\|_F \le \delta_A$), $\delta_A = 10^{-7}, 10^{-6}, \ldots, 10^{-1}$ for Type 1 matrices A, $\delta_A = 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}$ for Type 2 matrices A. The DLS estimate usually has no accurate digits compared with x if δ_A is taken to be larger.
- $y = \hat{x} + \frac{1}{\sqrt{40}}\delta_{\hat{x}}\|\hat{x}\|_2\,\mathtt{rand}(40, 1)$, $\hat{x}$ is the computed solution to the DLS problem (2.1), and $\delta_{\hat{x}} = 0, 10^{-7}, 10^{-6}, \ldots, 10^{-1}$.
- For each pair of δ_A and $\delta_{\hat{x}}$ and each type of matrix, we generated 1000 sample problems.

The solution $\hat{x}$ to the DLS problem satisfies (see [5])

$$\hat{x} = \frac{b^T b}{b^T A v_{_D}} v_{_D}, \tag{5.1}$$

where $v_{_D}$ is the right singular vector corresponding to the smallest singular value of $(I - bb^\dagger)A$. The equality (5.1) can also be obtained from [21, section 9], which suggests a way to compute $\hat{x}$. In our numerical tests we used the MATLAB built-in function `svd` to find $v_{_D}$ and then computed $\hat{x}$. To compute the asymptotic estimate $\tilde{\mu}_F(y)$, we first computed the QR factorization of B (see (4.6)) to find the Q-factor and then used (4.9).

In our numerical tests single precision was used to generate the data A, b, and y and to compute the DLS solution $\hat{x}$; both single precision and double precision were used to compute both sides of the inequality in (2.31) (where ΔA gives the minimum). The number of failures to satisfy the inequality for each case by single (S) and double (D) precision is reported in Table 5.1 for Type 1 matrices, and in Table 5.2 for Type 2 matrices. When δ_A is small or $\delta_{\hat{x}}$ is large, we see that the computed version of the inequality by single precision sometimes fails. In particular, for ill-conditioned Type 2 matrices, when $\delta_A = 10^{-7}$ or $\delta_{\hat{x}} = 10^{-1}$, the failure percentage is very large. However, the computed version of the inequality by double precision holds for these test cases. This shows that these failures were due to rounding errors in the single precision computed version of the inequality, and for these test

TABLE 5.1
Number of failures to satisfy the inequality (2.31) *out of* 1000 *samples for Type* 1.

			$\delta_{\hat{x}}$							
			0	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
δ_A	10^{-7}	S	9	2	4	2	1	1	2	1
		D	0	0	0	0	0	0	0	0
	10^{-6}	S	4	2	1	2	4	1	4	1
		D	0	0	0	0	0	0	0	0
	10^{-5}	S	0	0	0	0	0	4	3	0
		D	0	0	0	0	0	0	0	0
	10^{-4}	S	0	0	0	0	0	0	2	1
		D	0	0	0	0	0	0	0	0
	10^{-3}	S	0	0	0	0	0	0	0	0
		D	0	0	0	0	0	0	0	0
	10^{-2}	S	0	0	0	0	0	0	0	0
		D	0	0	0	0	0	0	0	0
	10^{-1}	S	0	0	0	0	0	0	0	0
		D	0	0	0	0	0	0	0	0

TABLE 5.2
Number of failures to satisfy the inequality (2.31) *out of* 1000 *samples for Type* 2.

			$\delta_{\hat{x}}$							
			0	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
δ_A	10^{-7}	S	718	725	726	749	965	997	1000	997
		D	0	0	0	0	0	0	0	0
	10^{-6}	S	2	0	1	0	107	970	1000	997
		D	0	0	0	0	0	0	0	0
	10^{-5}	S	0	0	0	0	0	120	945	995
		D	0	0	0	0	0	0	0	0
	10^{-4}	S	0	0	0	0	0	0	123	888
		D	0	0	0	0	0	0	0	0

cases the extended minimal backward error is actually the true minimal backward error. The reason that single precision rounding errors caused some tests to fail is almost certainly the following: in each failed case the gap between the smallest and the second smallest singular values of N_∞ was small, making the computation of the singular vector w_* (see (2.49)) inaccurate (see, e.g., [2, Theorem 1.2.8] or [8, Theorem 8.6.5] for perturbation results concerning the singular vectors). Indeed we noticed that, for the failed cases, w_* computed by single precision was very inaccurate compared with the one computed by double precision, leading to a large computational error in $\widehat{\Delta A}$, where this is needed for checking the inequality in (2.31).

In Figures 5.1 to 5.8 we give the plots corresponding to eight extreme cases in Tables 5.1 and 5.2 which exhibit $\mu_F(y)$ (as abscissa) versus $\mu_2^{\mathrm{lb}}(y)$ in (3.2), and $\mu_F(y)$ (as abscissa) versus $\tilde{\mu}_F(y)$ in (4.2) and (4.9), represented by the points $\cdot$ (blue) for $\mu_2^{\mathrm{lb}}(y)$, and $*$ (green) for $\tilde{\mu}_F(y)$. The diagonal (red) is plotted for reference. In these figures the above quantities were computed by double precision. But we can see no difference between these figures and the corresponding ones obtained by single precision.

From Figures 5.1, 5.2, 5.5, and 5.6, where each y is the computed DLS solution $\hat{x}$, we see that the minimal backward error $\mu_F(\hat{x}) \approx 10^{-7}$, which is close to the unit roundoff for single precision so that each computed $\hat{x}$ is a backward stable solution. It is interesting to see from Figures 5.3, 5.4, 5.7, and 5.8 that $\mu_F(y)$ is about one or two orders of magnitude smaller than $\delta_{\hat{x}}$. This phenomenon also holds for other test cases.

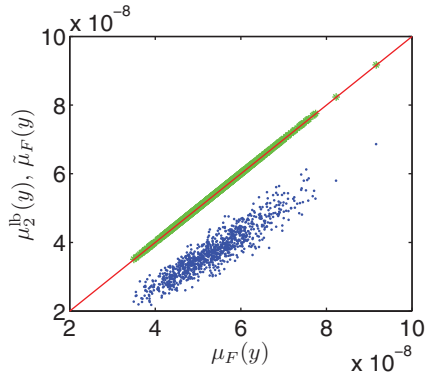


FIG. 5.1. *Type* 1 A, $\delta_A = 10^{-7}$, $\delta_{\hat{x}} = 0$.

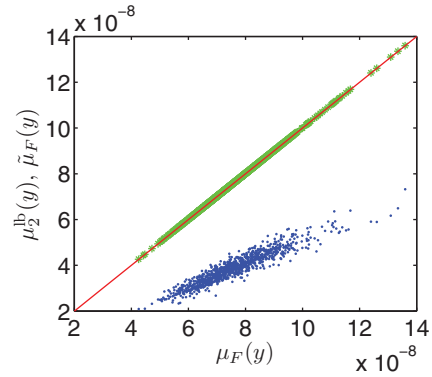


FIG. 5.2. *Type* 1 A, $\delta_A = 10^{-1}$, $\delta_{\hat{x}} = 0$.

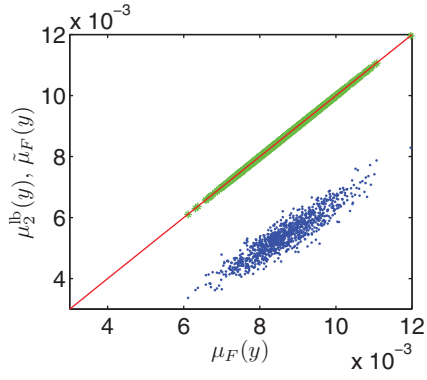


FIG. 5.3. *Type* 1 A, $\delta_A = 10^{-7}$, $\delta_{\hat{x}} = 10^{-1}$.

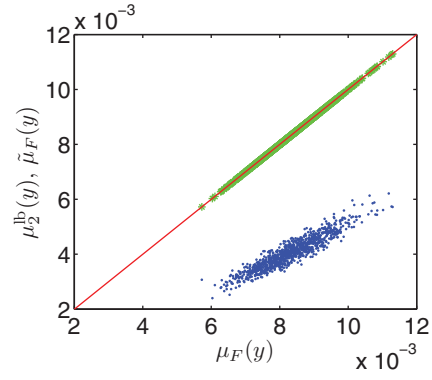


FIG. 5.4. *Type* 1 A, $\delta_A = 10^{-1}$, $\delta_{\hat{x}} = 10^{-1}$.

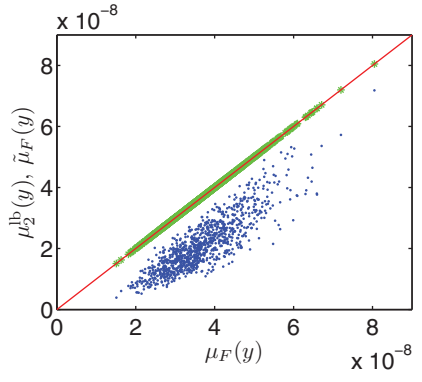


FIG. 5.5. *Type* 2 A, $\delta_A = 10^{-7}$, $\delta_{\hat{x}} = 0$.

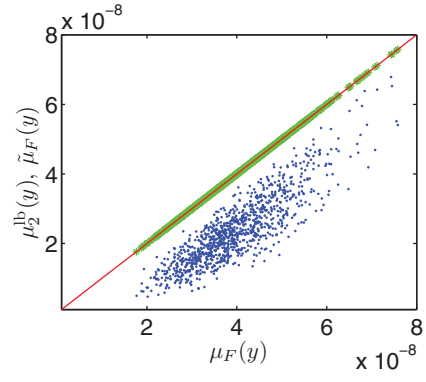


FIG. 5.6. *Type* 2 A, $\delta_A = 10^{-4}$, $\delta_{\hat{x}} = 0$.

All of these figures and the figures we did not display here indicate that the lower bound $\mu_2^{\rm lb}(y)$ is a reasonable approximation to the minimal backward error $\mu_F(y)$ in the sense that these two always had the same order of magnitude, although the case for Type 1 matrices is worse than the case for Type 2 matrices. We also see that the asymptotic estimate $\tilde{\mu}_F(y)$ is an excellent approximation to $\mu_F(y)$, even when y is not close to the DLS solution $\hat{x}$; see Figures 5.3, 5.4, 5.7, and 5.8, where $\delta_{\hat{x}} = 10^{-1}$.

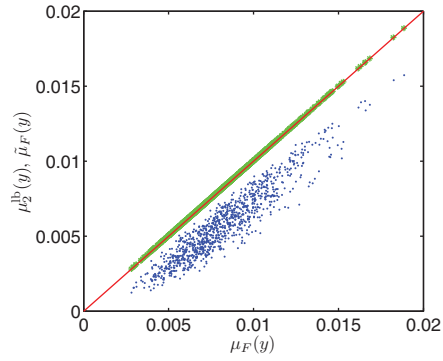


FIG. 5.7. *Type* 2 A, $\delta_A = 10^{-7}$, $\delta_{\hat{x}} = 10^{-1}$.

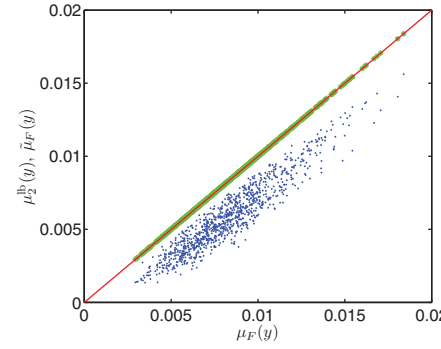


FIG. 5.8. *Type* 2 A, $\delta_A = 10^{-4}$, $\delta_{\hat{x}} = 10^{-1}$.

6. Summary and future work. For a given approximate solution y to the DLS problem (2.1), we first presented formulas (2.15) in Theorem 2.2 for an extended minimal backward error $\mu_F(y, \theta)$ for the case where backward perturbations in both A and b are allowed. Then by taking $\theta \to \infty$, we obtained the corresponding formulas (2.48) in Theorem 2.6 for an extended minimal backward error $\mu_F(y)$ for the case where only backward perturbations in A are allowed—this is the case we considered later in the paper. In theory $\mu_F(y)$ is a lower bound on the minimal backward error, but if the inequality in (2.31) is satisfied for the optimal perturbation $\widehat{\Delta A}$ given in (2.49), it is, in fact, the the minimal backward error. Our simulations showed that if y is a reasonable approximation to the exact DLS solution (that is, having a relative error in y of less than 10^{-1} for our test cases), then apparently the inequality in (2.31) holds in the absence of rounding errors in checking the inequality. Thus we believe in practice that $\mu_F(y)$ can usually be used as the minimal backward error. Since the formula (2.48) for $\mu_F(y)$ involves the minimum singular value of a matrix, it is expensive to compute directly. In order to overcome this problem, we derived a lower bound $\mu_2^{\mathrm{lb}}(y)$ (see (3.2)) and an asymptotic estimate $\tilde{\mu}_F(y)$ (see (4.6), (4.8), and (4.9)). These can be computed or estimated more efficiently. For our numerical test cases, $\mu_2^{\mathrm{lb}}(y)$ always had the same order of magnitude as $\mu_F(y)$, and $\tilde{\mu}_F(y)$ was an excellent approximation to $\mu_F(y)$. Since the computation of $\mu_2^{\mathrm{lb}}(y)$ is so inexpensive, it would seem to give a simple and effective indicator.

Several problems need to be investigated in the future. To check if the extended minimal backward error is the actual minimal backward error, we need an efficient and reliable way to test the inequality in (2.31) (or the inequality in (2.9) when perturbations in both A and b are allowed). The relationships between $\mu_F(y)$ and $\tilde{\mu}_F(y)$ needs to be studied further. We would also like to incorporate the results obtained in this paper to design effective stopping criteria for iterative algorithms for solving the DLS problem and extend the results here to total least squares problems (see [7] and [33]) and scaled total least squares problems (see [23] and [21]).

Acknowledgments. The first author is indebted to the second author and to Michael Saunders for their hospitality when he was on sabbatical at Stanford University where part of this research was done. He is also grateful to Zheng Su for helpful discussions. We also thank Pete Stewart and Nick Trefethen for improving our understanding of eigensystems of analytic Hermitian matrices. Two referees made many helpful suggestions, some of which were very demanding but led to necessary and sub-

stantial expansions of this paper. One referee also suggested the form of Theorem 2.8 in order to give a clearer exposition of our thinking.

REFERENCES

[1] M. Arioli, I. Duff, and D. Ruiz, *Stopping criteria for iterative solvers*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 138–144.

[2] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[3] X.-W. Chang, C. C. Paige, and D. Titley-Peloquin, *Characterizing matrices that are consistent with given solutions*, SIAM J. Matrix Anal. Appl., to appear.

[4] A. J. Cox and N. J. Higham, *Backward error bounds for constrained least squares problems*, BIT, 39 (1999), pp. 210–227.

[5] R. D. DeGroat and E. M. Dowling, *The data least squares problem and channel equalization*, IEEE Trans. Signal Process., 42 (1993), pp. 407–411.

[6] G. H. Golub and G. Meurant, *Matrices, moment and quadrature*, in Proceedings of the 15th Dundee Conference, 1993, Longman Scientific & Technical, D. F. Griffiths and G. A. Watson, eds., 1994, pp. 105–156.

[7] G. H. Golub and C. F. Van Loan, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore MD, 1996.

[9] J. F. Grcar, *Optimal Sensitivity Analysis of Linear Least Squares*, Technical report LBNL-52434, Lawrence Berkeley National Laboratory, Berkeley, CA, 2003.

[10] J. F. Grcar, *The optimum inverse problem of numerical error analysis*, Householder Symposium XVI, Champion, PA, 2005.

[11] J. F. Grcar, M. A. Saunders, and Z. Su, *Estimates of Optimal Backward Perturbations for Linear Least Squares Problems*, manuscript, 2004.

[12] M. Gu, *Backward perturbation bounds for linear least squares problems*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 363–372.

[13] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[14] D. J. Higham and N. J. Higham, *Backward error and condition of structured linear systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 162–175.

[15] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[16] T. Kato, *Perturbation Theory for Linear Operators*, 2nd ed., Springer-Verlag, New York, 1980.

[17] R. Karlson and B. Waldén, *Estimation backward perturbation bounds for the linear least squares problem*, BIT, 37 (1997), pp. 862–869.

[18] A. N. Malyshev, *Optimal backward perturbation bounds for the LSS problems*, BIT, 41 (2001), pp. 430–432.

[19] A. N. Malyshev and M. Sadkane, *Computation of optimal backward perturbation bounds for large sparse linear least squares problems*, BIT, 41 (2002), pp. 739–747.

[20] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[21] C. C. Paige and Z. Strakoš, *Scaled total least squares fundamentals*, Numer. Math., 91 (2002), pp. 117–146.

[22] C. C. Paige and Z. Strakoš, *Unifying least squares, total least squares and data least squares*, in Total Least Squares and Errors-in-Variables Modeling, S. van Huffel and P. Lemmerling, eds., Kluwer Academic Publishers, Dordrecht, 2002, pp. 25–34.

[23] B. D. Rao, *Unified treatment of LS, TLS and truncated SVD methods using a weighted TLS framework*, in Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modelling, S. Van Huffel, ed., SIAM Publications, Philadelphia, 1997, pp. 11–20.

[24] F. Rellich, *Perturbation Theory of Eigenvalue Problems*, Gordon and Breach, New York, 1969.

[25] J. L. Rigal and J. Gaches, *On the compatibility of a given solution with the data of a linear system*, J. ACM, 14 (1967), pp. 543–548.

[26] G. W. Stewart, *On the perturbation of pseudo-inverses, projections, and linear least squares problems*, SIAM Rev., 19 (1977), pp. 634–662.

[27] Z. Su, *Computational Methods for Least Squares Problems and Clinical Trials*, Ph.D. thesis, Stanford University, Palo Alto, CA, 2005.

[28] J.-G. Sun, *Optimal backward perturbation bounds for the linear least squares problem with multiple right-hand sides*, IMA J. Numer. Anal., 16 (1996), pp. 1–11.

[29] J.-G. Sun, *On optimal backward perturbation bounds for the linear least-squares problem*, BIT, 37 (1997), pp. 179–188.
[30] J.-G. Sun, *Bounds for the structured backward errors of Vandermonde systems*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 45–59.
[31] J.-G. Sun, *A note on backward errors for structured linear systems*, Numer. Linear Algebra Appl., 12 (2005), pp. 585–603.
[32] J.-G. Sun and Z. Sun, *Optimal backward perturbation bounds for underdetermined systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 393–402.
[33] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM Publications, Philadelphia, 1991.
[34] J. M. Varah, *Backward error estimates for Toeplitz systems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 408–417.
[35] B. Waldén, R. Karlson and J.-G. Sun, *Optimal backward perturbation bounds for the linear least squares problem*, Numer. Linear Algebra Appl., 2 (1995), pp. 271–286.
[36] H. Xiang and Y. Wei, *On normwise structured backward errors for saddle point systems*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 838-849.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1302–1319

OUT-OF-CORE IMPLEMENTATIONS OF CHOLESKY FACTORIZATION: LOOP-BASED VERSUS RECURSIVE ALGORITHMS*

NATACHA BÉREUX†

Abstract. We compare, in the same framework, out-of-core implementations of the Cholesky factorization algorithm. The candidate implementations are the classical blocked left-looking variant and a more recent recursive formulation. Both have been implemented for real positive definite matrices: the former in the parallel out-of-core linear algebra package (POOCLAPACK) library and the latter in the scalable out-of-core linear algebra computations (SOLAR) library. We perform a theoretical analysis of the amount of input/output (I/O) operations required by each variant. We consider alternatives for the left-looking algorithm: the one-tile and two-tiles approaches. We show that when main memory is restricted, the one-tile approach yields less I/O volume. We then show that the left-looking implementation requires less I/O volume than the recursive variant. We have implemented all for complex matrices, and we report on numerical experiments.

Key words. Cholesky factorization, out-of-core algorithms

AMS subject classifications. 15A23, 65F05, 65Y20

DOI. 10.1137/06067256X

1. Introduction.

Large, dense, and complex valued linear systems occur in industrial applications of the boundary element method (BEM). Using the disk through an out-of-core solver is an effective way to overcome memory limitations. The aim of this paper is to compare recursive and loop-based implementations of an out-of-core Cholesky solver.

1.1. Motivation.

The BEM, which is widely used in electromagnetic or acoustic scattering, consists of transforming the original scattering problem set in an unbounded domain into an integral equation set on the boundary of the scatterer [19]. In many situations, a symmetric formulation of the integral equation is preferred (such as the electric field integral equation in electromagnetism [6, 9]). The discretization of such an integral equation, e.g., by finite element methods leads to a linear system of equations:

$$A\,X = B, \tag{1.1}$$

where the coefficient matrix A of order N is dense, complex valued, symmetric but non-Hermitian, and the right-hand side B is a given complex valued matrix of size $N \times M$. It is observed that, for complex symmetric matrices issued from the discretization of boundary integral equations (BIE), pivoting is not required: thus, a Cholesky factorization can be used [7, 5]. Note that this special property is strongly linked to the underlying BIE problem. For general complex symmetric matrices, the Cholesky algorithm is unstable and may even breakdown.

For very large problems (e.g., $N \geq 100000$), direct solvers are too costly, and preconditioned iterative algorithms are preferred [8, 10]. For intermediate size problems where the matrix is too large to fit in-core, an out-of-core solver is very effective.

*Received by the editors October 17, 2006; accepted for publication (in revised form) by I. S. Dhillon May 21, 2008; published electronically October 16, 2008.
http://www.siam.org/journals/simax/30-4/67256.html

†CMAP, Ecole Polytechnique, CNRS, Route de Saclay 91128 Palaiseau, France. Current address: THEMIS Department, EDF R&D, 1 avenue du général de Gaulle, 92141 Clamart cedex, France (natacha.bereux@edf.fr).

1.2. State of the art. The limited size of the memory is a major bottleneck for solving large industrial problems, e.g., in electromagnetics. Therefore, out-of-core implementation, which performs linear algebra operations on matrices stored on the disk, is still an active resarch field. We refer to [24] for an extensive survey of trends in out-of-core algorithms and to [1] for recent developments related to sparse out-of-core parallel solvers. The Cholesky factorization is one of the classical linear algebra operations supported, for example, by linear algebra package (LAPACK) library [4].

The parallel extension of LAPACK, scalable linear algebra package (ScaLAPACK) provides a parallel out-of-core implementation of Cholesky factorization [11]: the matrix is partitioned into column panels and a left-looking variant of Cholesky factorization is chosen.

The parallel linear algebra package (PLAPACK) is a library infrastructure for coding linear algebra algorithms at a high level of abstraction [25]. Its parallel out-of-core linear algebra package extension (POOCLAPACK) provides a parallel out-of-core implementation of Cholesky factorization: the matrix is partitioned into tiles and is factorized by a left-looking variant [15]. The tile approach is more scalable than the panel approach used in ScaLAPACK. Two slightly different tile variants have been proposed in POOCLAPACK: the one-tile variant [26] and the two-tiles variant [21], depending on the number of full tiles allowed to reside in the memory simultaneously.

The scalable out-of-core linear algebra computations (SOLAR) library [22] is based on recursive formulations of linear algebra algorithms. It is designed to be portable across various architectures (personal, shared-memory, distributed memory computers) thanks to a portability layer, the matrix input output subroutines, which manages the data transfers to and from the disk. Matrices are partitioned into primary blocks. A primary block is the basic unit of the matrix, stored contiguously on the disk. In SOLAR's Cholesky factorization routine, the matrix is recursively split. At each level, the solution of a large system with a triangular matrix is computed, and a large symmetric rank-k update (SYRK) is performed through calls to the out-of-core basic linear algebra subroutines (BLAS). At the leaf level, the matrix is factorized by an in-core Cholesky routine (from LAPACK or ScaLAPACK). The recursion is stopped when the matrix size is equal to a predefined blocksize (for instance, when the matrix consists of 2×2 or 4×4 primary blocks). Experiments with the recursive Cholesky algorithm show that a significant amount of I/O operations is hidden by the computations. The recursive formulation of LU decomposition with partial pivoting drastically outperforms the left- and right-looking formulations.

Recursion is mainly thought to improve the locality of reference: in-core implementations perform fewer cache misses, and out-of-core implementations incur a smaller I/O volume. This feature is demonstrated for LU decomposition in [23]. Recursive data structures have also been developed in [3, 2]. Recently, a sparse Cholesky solver based on a recursive formulation of the Cholesky algorithm, at the sparse and at the dense level, combined with a recursive layout of the data has been proposed in [17]. This solver is part of the library of sparse linear solvers TAUCS. We refer to [13] for a complete survey on recursion as a key for designing high-performance linear algebra libraries.

1.3. Aim. We seek to compute the Cholesky decomposition of large dense matrices, stored on disk, on a computer with a limited amount of available memory. We focus on sequential algorithms. We survey several variants and memory layouts for computing this decomposition. Then, we propose a theoretical comparison of the re-

sulting implementations, substantiated by experiments, to find out the most effective implementation.

The paper is organized as follows. Section 2 describes two variants of the Cholesky algorithm: the left-looking variant and the recursive variant. Then, out-of-core implementations of these variants are detailed in section 3. We describe, in particular, the one-tile and the two-tiles approaches, which correspond to different memory layout. These out-of-core implementations rely on out-of-core BLAS, which are described in section 4. A theoretical analysis of the data volume from the disk to the memory is performed in section 5 for the left-looking variant and in section 6 for the recursive variant. Experiments are conducted in section 7. Our conclusions are presented in section 8.

2. The Cholesky factorization. We recall the definition of the Cholesky decomposition of a symmetric matrix A, along with assumptions and notations used throughout the paper. Then, two algorithms for the computation of the Cholesky factor L are reviewed: a loop-based algorithm (the left-looking Cholesky algorithm) and a recursive algorithm. A loop-based algorithm is based on nested do-loops, whereas a recursive algorithm is based on a recursive splitting of the matrix.

2.1. Definitions and notations.

2.1.1. Definition. Let A be a symmetric matrix, admitting the decomposition:

$$A = L\,L^T, \tag{2.1}$$

where L is a lower triangular matrix. This decomposition exists and may be computed in a stable way by the so-called Cholesky algorithm when A is a real definite positive matrix, or when A is a complex symmetric matrix encountered by the BEM [5, 7].

In the following, chol denotes a generic routine for the computation of the lower triangular Cholesky factor. Depending on the context, chol stands for a variant of the algorithm or for an optimized computational kernel.

2.1.2. Partitioned algorithms. All variants of the Cholesky algorithm considered in this paper are partitioned algorithms. The matrices A and L are partitioned into tiles. Basic operations are matrix-matrix operations on these tiles, such as multiply, add.... These operations are provided by the level 3 BLAS and are known to enjoy a high level of data-reuse [12]. Therefore, partitioned algorithms built on calls to level 3 BLAS benefit from this high level of data-reuse, which is critical for an efficient out-of-core implementation [24].

2.1.3. Notations. The matrix A is a real or complex symmetric matrix of size $n \times n$, partitioned into $p \times p$ tiles of size $t \times t$ each. The dimension p, in tiles, is approximately equal to n/t. The dimension of A, in tiles, is sometimes denoted by $p(A)$, to avoid an ambiguity.

2.2. A loop-based algorithm: The left-looking variant. We follow the formalism embraced in [20] to recap the left-looking variant of the Cholesky algorithm. Let A and L be partitioned into quadrants with square diagonal blocks:

$$A = \begin{pmatrix} A_{TL} & \star \\ A_{BL} & A_{BR} \end{pmatrix}, \quad L = \begin{pmatrix} L_{TL} & 0 \\ L_{BL} & L_{BR} \end{pmatrix}.$$

Here, A_{TL} and L_{TL} are square, and the diagonal blocks of L, L_{TL}, and L_{BL} are lower triangular. The $\star$ indicates that the corresponding part of the matrix is not referenced (due to symmetry).

Algorithm: $[A] := \text{LLT}(A)$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & \star \\ \hline A_{BL} & A_{BR} \end{array}\right)$
where A_{TL} is 0×0

until A_{BR} is 0×0 **do**
Repartition

$$\left(\begin{array}{c|c} A_{TL} & \star \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & \star & \star \\ \hline A_{10} & A_{11} & \star \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

where A_{11} is $b \times b$

$$\begin{array}{ll} \mathbf{1} & A_{11} := A_{11} - A_{10}\, A_{10}^T \\ \mathbf{2} & A_{11} := L_{11} = \text{CHOL}\,(A_{11}) \\ \mathbf{3} & A_{21} := A_{21} - A_{20}\, A_{10}^T \\ \mathbf{4} & A_{21} := A_{21}\, L_{11}^{-T} \end{array}$$

Continue with

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & \star & \star \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

enddo

FIG. 2.1. *Loop-based, partitioned, left-looking variant of the Cholesky algorithm.*

Using the partitionings in (2.1), we obtain

$$\begin{pmatrix} A_{TL} & \star \\ A_{BL} & A_{BR} \end{pmatrix} = \begin{pmatrix} L_{TL}\, L_{TL}^T & \star \\ L_{BL}\, L_{TL}^T & L_{BL}\, L_{BL}^T + L_{BR}\, L_{BR}^T \end{pmatrix}. \tag{2.2}$$

This equality must hold when the factorization is completed. A possible choice for the content of A at an intermediate step of the factorization is

$$A := \begin{pmatrix} L_{TL} & \star \\ A_{BL}\, L_{TL}^{-T} & A_{BR} \end{pmatrix}. \tag{2.3}$$

The bottom-right part of the matrix A_{BR} is not changed. The loop-invariant (2.3) yields the so-called left-looking variant, described in Figure 2.1.

2.3. A recursive algorithm. Observe that

$$\mathsf{chol}\,(A) := \begin{pmatrix} L_{TL} & \star \\ L_{BL} & L_{BR} \end{pmatrix},$$

where

$$\begin{aligned} L_{TL} &= \mathsf{chol}\,(A_{TL}), \\ L_{BL} &= A_{BL}\, L_{TL}, \\ L_{BR} &= \mathsf{chol}\,(A_{BR} - L_{BL}\, L_{BL}^T). \end{aligned}$$

The diagonal blocks are computed by recursive calls to the chol decomposition, yielding a natural recursive description of Cholesky decomposition [13], based on a recursive splitting of A, until the subblocks consist of a single tile. The main steps of this variant are described in Figure 2.2. In [23], Toledo shows that a recursive formulation of LU decomposition improves the locality of reference. The recursive algorithm is thus not only more concise but also more efficient than the classical right-looking variant of LAPACK. Recursive blocked algorithms have been introduced for several dense and sparse linear algebra operations, e.g., Cholesky decomposition [2, 16, 3, 17, 22].

Algorithm: $[A] := \text{RLLT}(A)$

if $p(A) > 1$ **do**

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & \star \\ \hline A_{BL} & A_{BR} \end{array}\right)$

where $p(A_{TL}) = p(A)/2$

1 $A_{TL} := L_{TL} = \text{RLLT}(A_{TL})$

2 $A_{BL} := A_{BL}\, L_{TL}^{-T}$

3 $A_{BR} := A_{BR} - A_{BL} A_{BL}^{T}$

4 $A_{BR} := L_{BR} = \text{RLLT}(A_{BR})$

else

$A := L = \text{CHOL}(A)$

endif

FIG. 2.2. *A recursive variant of the Cholesky algorithm. The dimension of matrix A, in tiles, is denoted by* $p(A)$.

3. Out-of-core implementations. From now on, the matrix is assumed to reside on disk. We describe the data layout and then detail the implementation of a left-looking variant and of a recursive variant of the Cholesky algorithm. Two implementations of a Cholesky left-looking variant are sketched: a two-tiles and a one-tile, depending on the number of tiles (respectively two and one) allowed to reside in memory simultaneously.

3.1. Data layout. We follow the tile approach already implemented in POOCLAPACK [15, 26].

3.1.1. Matrix tiling. We use two types of partitioning of the matrix:

- a **recursive** partitioning: the interval $[1, n]$ is recursively halved until the size of the subintervals is less than or equal to t.
- an **arithmetic** partitioning: the interval $[1, n]$ is partitioned into subintervals of size t, except possibly for the first subinterval.

Examples of partitioning are shown in Figure 3.1. In the arithmetic partitioning, the larger subintervals are the last ones: the tiles updated by more computation lay

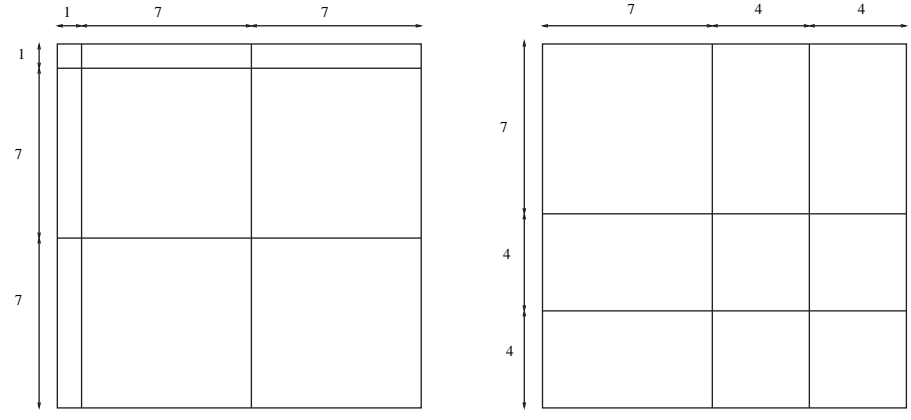


FIG. 3.1. *A matrix of size* 15×15 *is partitioned using an arithmetic partitioning (on the left) and using a recursive partitioning (on the right). In both cases, the tile size* t *is equal to* 7*: it represents the default block size in the arithmetic partitioning and the maximum size of the leaves in the recursion tree for the recursive partitioning.*

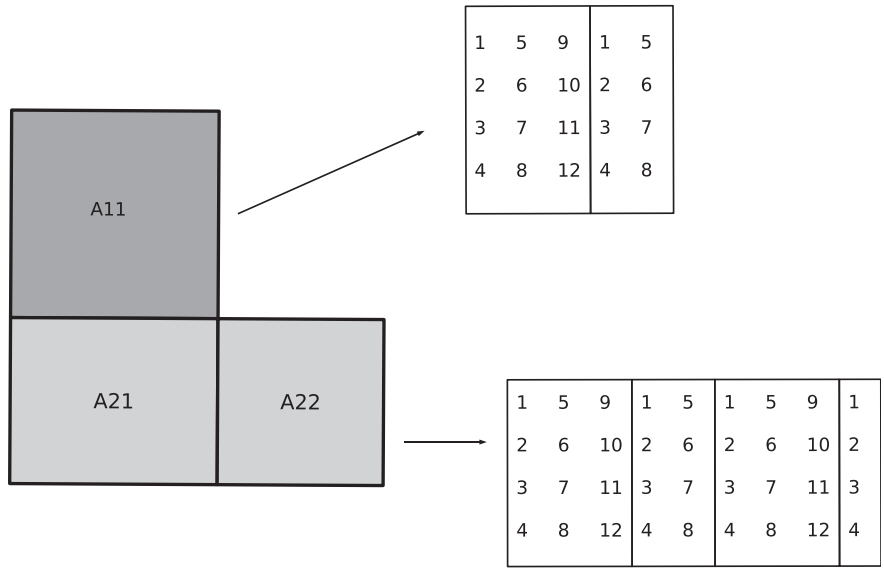


FIG. 3.2. *An example of a matrix out-of-core storage. The matrix of size* 9×9 *is recursively partitioned into tiles of size* $t \times t$*, with* $t = 5$*. The matrix is stored on the disk by block-rows. Hence, two direct-access files are needed: one contains the tile* A_{11}*, and the other the tiles* A_{21} *and* A_{22}*. Each tile is stored as a collection of records, where each record corresponds to a block of size* $t \times b$ *(*$b = 3$*), stored in column-major order.*

indeed at the bottom-right part of the matrix, and it is better to have them as large as possible to increase the operations-to-I/O ratio.

The recursive variants of Cholesky algorithms are based on a recursive partitioning of the matrix. The left-looking variants are usually based on an arithmetic partitioning.

The size of the main memory constrains the size of the tile, but p can be very large (it is only constrained by the size of the disk). Hence, this approach is well-adapted to large out-of-core matrices.

Another choice is made in the out-of-core parallel implementation of Cholesky factorization in ScaLAPACK: the matrix is partitioned into slabs of columns (possibly of variable width), instead of tiles. But this approach is known to lack scalability [24, 18].

3.1.2. Storage details. The partitioned matrix is decomposed in block-rows of height t. Each row is stored tile by tile. We will assume that each tile itself is stored in column-major order.

In practical, each block-row is stored as a collection of records in a direct-access file. In this file, each record corresponds to a primary block, stored contiguously on disk. The storage is illustrated by Figure 3.2. We distinguish between a block and a tile. A [illegible] corresponds to a physical record in the direct-access file storing a block-row of the matrix: it is the smallest unit in which the matrix is stored. A [illegible] corresponds to an element of the logical partitioning of the matrix. A tile is split on disk into several blocks of size $t \times b$, with $b \leq t$. In the following, blocks are called [illegible], since they generally have fewer columns than rows. Note that if $b = t$, a tile is identical to a block.

3.2. I/O scheme. The I/O scheme is a simple synchronous scheme, based on the standard read/write subroutines. Narrow blocks are stored as records in direct-

access files. These records are explicitly read from (and written back to) the disk when needed. No prefetching (overlapping of I/O with computation) is employed.

3.3. Implementations of the left-looking variant. We describe two implementations of the left-looking variant for the computation of the Cholesky factor of matrix A:

- In the [illegible] approach, two tiles are allowed to reside in-core simultaneously [15].
- In the [illegible] approach, there is only one tile in-core [26]. Therefore, this tile may be larger, and the implementation benefits from a better ratio between computations and I/O operations.

In both approaches, two extra buffers are needed for the in-core storage of two narrow blocks.

3.3.1. Two-tiles approach. We strictly follow the sequential implementation stated in [15]. Two tiles simultaneously reside in the main memory. One contains the diagonal tile A_{11}, and the other contains the current tile of column A_{21}. We recap the main steps and refer to [15] and [21] for further details:

- [illegible] 1: Tile A_{11} is read from disk and updated as $A_{11} := A_{11} - A_{10}A_{10}^T$ by a sequence of narrow out-of-core symmetric rank-k updates (routine OOC_TILE_SYRK).
- [illegible] 2: Tile A_{11} is factored by an in-core factorization routine. Then, tile A_{11} is written back on disk, but the corresponding in-core buffer is kept in memory.
- [illegible] 3 [illegible] 4: The update of block-column A_{21}, $A_{21} := (A_{21} - A_{20}\,A_{10}^T)L_{11}^{-T}$ is performed tile by tile. Each tile of A_{21} is brought in turn into memory, updated by a sequence of narrow out-of-core multiply and adds (routine OOC_TILE_GEMM), and next, by solving (in-core) a multiple right-hand side triangular system of matrix A_{11}^T. Then, the current tile of A_{21} is written back on disk, the buffer containing A_{11} is flushed, and the next tile is read.

To optimize data transfers, tile A_{11} remains in-core for Steps 1 to 4. Therefore, we define mixed BLAS OOC_TILE_SYRK and OOC_TILE_GEMM, whose arguments are an in-core tile and out-of-core tiles:

- OOC_TILE_SYRK overwrites the (in-core) tile C of size $t \times t$ with $C - A\,A^T$, where the tile A is stored on disk.
- OOC_TILE_GEMM overwrites the (in-core) tile C of size $t \times t$ with $C + A\,B^T$, where A and B are out-of-core tiles.

These operations can be implemented as sequences of operations on narrow matrices, as advocated in [15]. The narrow block technique analyzed in section 5.5 allows us to increase the ratio of computation over I/O operations.

3.3.2. One-tile approach. The motivation of the one-tile approach introduced in [26] is to increase the size of the current tile, which is read and written. We briefly state the differences with the two-tiles implementation, for each step of the algorithm:

- Steps 1 and 2 are not changed.
- Tile A_{11} no longer remains in-core after Step 2. It is replaced in-core by the current tile of A_{21}. Then the routine OOC_TILE_GEMM is used to update A_{21}.
- For Step 4, a new mixed BLAS is needed for the solution of the triangular system with matrix A_{11}: OOC_TILE_TRSM. This routine overwrites the (in-core) tile A_{21} with the solution X of $X\,A_{11}^T = A_{21}$, where the triangular tile A_{11} is stored on disk. Tile A_{11} is read narrow block by narrow block in one of the two extra buffers provided.

This implementation uses a larger current tile and benefits from a better flops-to-I/O-operations ratio for the mixed BLAS, as we will see. Nevertheless, tile A_{11} has to be read twice.

3.4. Implementations of the recursive variant. Let us now consider the out-of-core implementation of the recursive factorization sketched in Algorithm 2.2. A similar implementation is reported in [22], in the framework of the SOLAR project.

3.4.1. Basic implementation. The main features of the factorization are the following:

- [illegible] of the matrix until the matrix consists of a single tile.
- [illegible] 1: At the leaf level, tile A_{11} is brought into memory, factored by an in-core Cholesky routine, and written back to the disk. At another level, the factorization is called recursively.
- [illegible] 2 consists in solving a triangular linear system with matrix L_{TL}. At the leaf level, this operation is performed by an in-core routine (for instance, the BLAS routine TRSM), after reading the tile L_{TL}. If the matrices are larger, an out-of-core routine OOC_TRSM is called.
- [illegible] 3 is a symmetric update operation. It is performed either by calling an in-core routine (for instance, the BLAS routine SYRK) or an out-of-core routine OOC_SYRK.

The recursive implementation calls two out-of-core BLAS, whose arguments are all out-of-core matrices:

- OOC_SYRK overwrites the out-of-core matrix C of size $n \times n$ with $C - A\,A^T$, where A is an out-of-core matrix of size $n \times k$. If $n = t$, A and C are brought into memory, and an in-core routine is called.
- OOC_TRSM overwrites the out-of-core matrix B of size $m \times n$ by the solution X of $X\,A^T = B$, where A is a triangular out-of-core matrix of size $n \times n$. When $m = n = t$, both arguments are brought into memory, and an in-core routine is called.

So, out-of-core routines switch to in-core routine as soon as the arguments are matrices consisting of a single tile.

3.4.2. Optimization: A (two-tiles) hybrid variant. Observe the factorization of a matrix of 2×2 tiles:

$$A = \begin{pmatrix} A_{11} & \star \\ A_{21} & A_{22} \end{pmatrix}.$$

The recursive procedure starts by splitting the matrix. Then the computation with a two-tiles implementation requires reading 5 tiles and writing 3 tiles. The following (also two-tiles) schedule avoids reading a tile immediately after it has been written:

- read, factorize A_{11};
- read, update A_{21};
- write A_{11} on disk and read A_{22},
- update A_{22}, write A_{21}, factorize and write A_{22}.

It involves only reading and writing 3 tiles. So it is proposed to stop the recursion when the matrix has 2×2 tiles and to switch to this schedule. The total gain in I/O volume is evaluated in section 6.5.

4. Out-of-core BLAS. The recursive schedule of Cholesky factorization stated in section 3.4 calls out-of-core BLAS, for the out-of-core computation of the following:

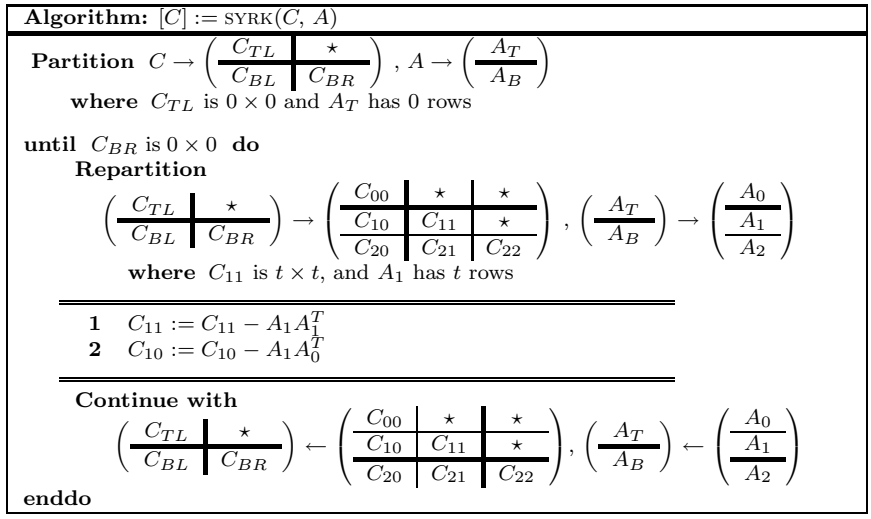


FIG. 4.1. *Blocked, loop-based algorithm for the computation of the SYRK, $C := C - A\,A^T$.*

- A symmetric rank-k update, OOC_SYRK, $C - A\,A^T$, where A is a $n \times k$ matrix and C is a $k \times k$ matrix (see Figure 4.1).
- The solution X of $X\,A^T = B$, OOC_TRSM, where A is a triangular matrix of size $n \times n$ and X overwrites B (of size $m \times n$); see Figure 4.2.

Out-of-core BLAS may be based on recursive algorithms or on loop-based algorithms. Some tests were conducted with recursive implementations but the performance was quite disappointing. So, we do not take the recursive option here and restrict (as in [22]) to loop-based algorithms. One-tile and two-tiles implementations of the out-of-core BLAS are needed as building blocks for the one-tile and two-tiles implementations of the recursive Cholesky algorithm. We briefly recall these implementations for OOC_SYRK and OOC_TRSM (see Chapter 8 and Chapter 16 of [14]).

4.1. Out-of-core SYRK. We recap the (one-tile) block down-right moving variant of the symmetric rank-k update SYRK:

- Tile C_{11} is brought into the main memory.
- [illegible] 1 is performed by applying a sequence of narrow rank-k updates through a call to OOC_TILE_SYRK. Then, C_{11} is written back on the disk.
- [illegible] 2 is performed tile by tile. Each tile of C_{10} is read, updated by a call to OOC_TILE_GEMM, and written back on disk.

4.2. Out-of-core TRSM. The solution of $X\,L^T = B$ is computed by the blocked right-moving variant of the triangular solver TRSM. The main steps of the out-of-core two-tiles implementation are

- Tile L_{11} is brought into memory.
- Column B_1 is replaced tile by tile by the corresponding column of tiles of the solution. The current tile of B_1 is read, updated by calling the routine OOC_TILE_GEMM. Then, the linear system with matrix L_{11}^T is solved in-core, and the current tile of B_1 is written back to the disk.

In the one-tile implementation, the tile L_{11} does not stay in-core. For every tile of B_1, the mixed BLAS OOC_TILE_TRSM is called instead of the in-core TRSM.

Algorithm: $[B] := \text{TRSM}(L, B)$

Partition $L \rightarrow \left(\begin{array}{c|c} L_{TL} & \star \\ \hline L_{BL} & L_{BR} \end{array}\right)$, $B \rightarrow \left(\begin{array}{c|c} B_L & B_R \end{array}\right)$

where L_{TL} is 0×0 and B_L has 0 columns

until L_{BR} is 0×0 **do**

Repartition

$$\left(\begin{array}{c|c} L_{TL} & \star \\ \hline L_{BL} & L_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} L_{00} & \star & \star \\ \hline L_{10} & L_{11} & \star \\ \hline L_{20} & L_{21} & L_{22} \end{array}\right), \left(\begin{array}{c|c} B_T & B_B \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} B_0 & B_1 & B_2 \end{array}\right)$$

where L_{11} is $t \times t$, and B_1 has t columns

$B_1 := (B_1 - B_0 L_{10}^T)\, L_{11}^{-T}$

Continue with

$$\left(\begin{array}{c|c} L_{TL} & \star \\ \hline L_{BL} & L_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} L_{00} & \star & \star \\ \hline L_{10} & L_{11} & \star \\ \hline L_{20} & L_{21} & L_{22} \end{array}\right), \left(\begin{array}{c|c} B_L & B_R \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} B_0 & B_1 & B_2 \end{array}\right)$$

enddo

FIG. 4.2. *Blocked, loop-based algorithm for the computation of the solution of the linear system* $X\,L^T = B$*, with a lower triangular matrix* L*, where* X *overwrites* B*.*

5. Analysis of the I/O costs of the left-looking variant. This section presents a theoretical analysis of data transfers between the disk and the main memory in left-looking Cholesky algorithms.

We compare the number of tiles read and written by the two-tiles and one-tile implementations of the left-looking Cholesky variant. Assuming that the main memory is limited, we obtain a simple criterion for choosing one or the other approach, when one wishes to minimize the I/O volume during the factorization. We then investigate the impact of using narrow blocks in terms of I/Os.

5.1. Notations. We follow a systematic naming scheme. The number of tiles read (respectively written) during the factorization of the matrix A is denoted by Tr (respectively, Tw). The number of matrix elements read (respectively written) is similarly denoted by Nr (respectively, Nw). A subscript characterizes the variant (LLT for the left-looking variant), and a superscript characterizes the implementation ((1T) for the one-tile implementation and (2T) for the two-tiles implementation). Their dependence with respect to (p, n, t, b) is made explicit when necessary.

For the sake of simplicity, we assume that n is a multiple of t, and thus $p = p(n, t) = n/t$.

The main memory is limited to M words. A narrow block is a subtile of size $t \times b$.

5.2. Two-tiles approach. We analyze the data transfers performed by the schedule described in section 3.3.1. Suppose that $k-1$ columns of the tiles of A have already been factored. Matrix A is partitioned as

$$\left(\begin{array}{c|c|c} A_{00} & \star & \star \\ \hline A_{10} & A_{11} & \star \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right), \tag{5.1}$$

where A_{11} is a tile and A_{22} is a matrix of $(p-k) \times (p-k)$ tiles.
During the kth iteration, the following I/O operations are performed:

- Read A_{11}, A_{10} (k tiles).
- Write A_{11} (1 tile).
- Read A_{21} tile by tile ($p-k$ tiles).

- For each tile of A_{21}, read A_{10} and $k-1$ tiles of A_{20} ($2 \times (k-1)$ tiles).
- Write A_{21} ($p-k$ tiles).

Hence, the total number of tiles read $\mathrm{Tr}_{\mathrm{LLT}}^{(2\mathrm{T})}(p)$ is

$$\mathrm{Tr}_{\mathrm{LLT}}^{(2\mathrm{T})}(p) = \sum_{k=1}^{p} p + 2\,(k-1)\;(p-k) = \frac{p}{3}\;\left(p^2+2\right).$$

The total number of tiles written $\mathrm{Tw}_{\mathrm{LLT}}^{(2\mathrm{T})}$ is

$$\mathrm{Tw}_{\mathrm{LLT}}^{(2\mathrm{T})}(p) = \sum_{k=1}^{p} k = \frac{p}{2}\,(p+1).$$

5.3. One-tile approach. We examine the number of tiles read and written during the kth iteration of the one-tile approach described in section 3.3.2. Matrix A is partitioned as in (5.1).

The tile A_{11} does not stay in memory for the whole iteration: it has to be read for each tile of A_{21}, which results in $p-k$ extra tiles to read. In the end, the number of tiles read $\mathrm{Tr}_{\mathrm{LLT}}^{(1\mathrm{T})}(p)$ is

$$\mathrm{Tr}_{\mathrm{LLT}}^{(1\mathrm{T})}(p) = \mathrm{Tr}_{\mathrm{LLT}}^{(2\mathrm{T})}(p) + \sum_{k=1}^{p}(p-k) = \frac{p}{6}\;\left(2p^2+3p+1\right),$$

and the number of tiles written $\mathrm{Tw}_{\mathrm{LLT}}^{(1\mathrm{T})}(p)$ is

$$\mathrm{Tw}_{\mathrm{LLT}}^{(1\mathrm{T})}(p) = \mathrm{Tw}_{\mathrm{LLT}}^{(2\mathrm{T})}(p) = \frac{p}{2}\;(p+1).$$

5.4. One-tile versus two-tiles implementation. We compare the two-tiles and one-tile implementations from the point of view of I/O volume.

The number of matrix elements read by the two-tiles and one-tile implementations, respectively, are given by,

$$\mathrm{Nr}_{\mathrm{LLT}}^{(2\mathrm{T})}(n,t) = \mathrm{Tr}_{\mathrm{LLT}}^{(2\mathrm{T})}\left(p=\frac{n}{t}\right) \times t^2 = \frac{n}{3\,t}\;\left(\left(\frac{n}{t}\right)^2+2\right), \tag{5.2}$$

$$\mathrm{Nr}_{\mathrm{LLT}}^{(1\mathrm{T})}(n,t) = \mathrm{Tr}_{\mathrm{LLT}}^{(1\mathrm{T})}\left(p=\frac{n}{t}\right) \times t^2 = \frac{n}{6\,t}\;\left(2\left(\frac{n}{t}\right)^2+3\left(\frac{n}{t}\right)+1\right). \tag{5.3}$$

The available main memory is partitioned into two-tiles of size $t=\sqrt{M/2}$ or one tile of size $t=\sqrt{M}$, respectively. Here, we neglect the two extra buffers needed for the storage in-core of two narrow blocks. The number of terms read with the two-tiles and the one-tile approach are equal to

$$\mathrm{Nr}_{\mathrm{LLT}}^{(2\mathrm{T})}\left(n,t=\sqrt{M/2}\right) = \frac{n\,\sqrt{M}}{3}\;\sqrt{2}\;\left(\frac{n^2}{M}+1\right),$$

$$\mathrm{Nr}_{\mathrm{LLT}}^{(1\mathrm{T})}\left(n,t=\sqrt{M}\right) = \frac{n\,\sqrt{M}}{3}\;\left(\frac{n^2}{M}+\frac{3}{2}\frac{n}{\sqrt{M}}+1\right).$$

For large values of n, the one-tile implementation requires less I/O operations than the two-tiles.

5.5. Narrow blocks. Some basic linear algebra operations, e.g., SYRK and matrix multiply and add (GEMM) can be implemented using the narrow block technique. The matrix operands that are left unchanged on exit of these operations are partitioned into narrow blocks. Hence, a GEMM

$$C \leftarrow C - AB^T$$

is implemented as a sequence of narrow multiply and adds: $C - \sum_i A_i B_i^T$, where A_i and B_i are narrow subblocks of A and B of width b.

This scheduling does not change the I/O volume for a fixed tile size. It improves the I/O volume when the memory size is limited. A two-tiles implementation actually uses two tiles and two narrow blocks in-core, that is, $2\,tb + 2\,t^2$ words. If b is small, t can increase up to the asymptotic value $\sqrt{M/2}$. On the opposite, if $b = t$, the maximum tile size is $t = \sqrt{M/4}$. The ratio of the number of matrix elements read in both situations is

$$\frac{\mathrm{Nr}_{\mathrm{LLT}}^{(2\mathrm{T})}\left(n, t = \sqrt{M/2}\right)}{\mathrm{Nr}_{\mathrm{LLT}}^{(2\mathrm{T})}\left(n, t = \sqrt{M/4}\right)} \simeq \frac{1}{\sqrt{2}}, \tag{5.4}$$

which is a quantitative estimate of the potential gain when using narrow blocks. The numerical experiments reported in section 7.1 confirm these asymptotic I/O savings.

A similar analysis is conducted for the one-tile variant. One tile and two narrow blocks reside in memory, consisting of $t^2 + 2tb$ words of memory. If b is small, the tile size goes to $\sqrt{M/2}$, whereas if $b = t$, t equals $\sqrt{M/3}$. The ratio of the number of elements read is asymptotically

$$\frac{\mathrm{Nr}_{\mathrm{LLT}}^{(1\mathrm{T})}\left(n, t = \sqrt{M/2}\right)}{\mathrm{Nr}_{\mathrm{LLT}}^{(1\mathrm{T})}\left(n, t = \sqrt{M/3}\right)} \simeq \frac{\sqrt{3}}{2\sqrt{2}}. \tag{5.5}$$

6. Analysis of the I/O costs of the recursive variant. In this section, we analyze the memory behavior of the two-tiles implementation of the recursive variant of the Cholesky factorization. We first estimate the number of I/Os for the out-of-core BLAS OOC_TRSM and OOC_SYRK. Then, in section 6.3, we combine these estimates to obtain the number of tiles read and written by the recursive variant of Cholesky factorization, when the number of tiles of the matrix A is a power of two.

6.1. Notations. Notations are similar to the notations introduced in section 5.1.

The subscripts RLLT, HLLT, SYRK, and TRSM, respectively, refer to the recursive variant of the Cholesky algorithm, the hybrid variant proposed in section 3.4.2, and the out-of-core BLAS OOC_SYRK and OOC_TRSM. Unless stated, implementations are two-tiles, and superscript (2T) is omitted.

6.2. Analysis of the out-of-core BLAS. We analyze the data transfer performed by an out-of-core (two-tiles) implementation of OOC_SYRK and OOC_TRSM. The operands of these routines are out-of-core matrices partitioned into tiles of size $t \times t$.

6.2.1. Out-of-core SYRK. Let A be a symmetric matrix of $p \times k$ tiles, and let C be a matrix of $p \times p$ tiles. The number of tiles read during the computation of $C - AA^T$ (see Figure 4.1) is

$$\mathrm{Tr}_{\mathrm{SYRK}}(p,k) = \sum_{j=1}^{p} 1 + k + (j-1)(2k+1) = k\,p^2 + \frac{p}{2}\,(p+1)\,. \tag{6.1}$$

The number of tiles written $\mathrm{Tw}_{\mathrm{SYRK}}(p,k)$ is

$$\mathrm{Tw}_{\mathrm{SYRK}}(p,k) = \frac{p}{2}\,(p+1). \tag{6.2}$$

6.2.2. Out-of-core TRSM. Let A be a lower triangular matrix of $p \times p$ tiles, and let B be a matrix of $m \times p$ tiles. The number of tiles read during the computation of the solution of $X\,A^T = B$ (see Figure 4.2) is

$$\mathrm{Tr}_{\mathrm{TRSM}}(m,p) = \sum_{k=1}^{p} 1 + m\,(1 + 2(k-1)) = m\,p^2 + p. \tag{6.3}$$

The number of tiles written $\mathrm{Tw}_{\mathrm{TRSM}}(m,p)$ is

$$\mathrm{Tw}_{\mathrm{TRSM}}(m,p) = \sum_{k=1}^{p} m = mp. \tag{6.4}$$

6.3. Analysis of the recursive variant. This subsection presents an analysis of the data transfers associated to the recursive variant of Cholesky factorization (see Figure 2.2). We assume that $p = 2^l$ for some integer l.

Let us compute $\mathrm{Tr}_{\mathrm{RLLT}}(2^l)$ and $\mathrm{Tw}_{\mathrm{RLLT}}(2^l)$ by induction.

If $k = 0$, the matrix A is reduced to a single tile, and the factorization is computed by the in-core Cholesky factorization routine. Hence,

$$\mathrm{Tr}_{\mathrm{RLLT}}(1) = 1, \quad \mathrm{Tw}_{\mathrm{RLLT}}(1) = 1.$$

Let $k \geq 1$. The number of tiles read at level k satisfies the following recurrence:

$$\mathrm{Tr}_{\mathrm{RLLT}}\left(2^k\right) = 2\,\mathrm{Tr}_{\mathrm{RLLT}}\left(2^{k-1}\right) + \mathrm{Tr}_{\mathrm{TRSM}}\left(2^{k-1}, 2^{k-1}\right) + \mathrm{Tr}_{\mathrm{SYRK}}\left(2^{k-1}, 2^{k-1}\right). \tag{6.5}$$

Hence, we have, for the factorization of a matrix of 2^l tiles,

$$\begin{aligned} \mathrm{Tr}_{\mathrm{RLLT}}\left(2^l\right) = 2^l\,\mathrm{Tr}_{\mathrm{RLLT}}(1) &+ \sum_{k=1}^{l} 2^{k-1}\,\mathrm{Tr}_{\mathrm{TRSM}}\left(2^{l-k}, 2^{l-k}\right) \\ &+ \sum_{k=1}^{l} 2^{k-1}\,\mathrm{Tr}_{\mathrm{SYRK}}\left(2^{l-k}, 2^{l-k}\right). \end{aligned} \tag{6.6}$$

Hence, using (6.1) and (6.3) in (6.6), we obtain the number of tiles read:

$$\mathrm{Tr}_{\mathrm{RLLT}}(2^l) = \frac{2^{3l}}{3} + \frac{2^{2l}}{4} + 2^l \left(\frac{3l}{4} + \frac{5}{12}\right). \tag{6.7}$$

The number of tiles written also satisfies relation (6.5). Using (6.2) and (6.4), we obtain

$$\mathrm{Tw}_{\mathrm{RLLT}}(2^l) = \frac{3}{4}\,2^{2l} + 2^l \left(\frac{l+1}{4}\right). \tag{6.8}$$

If p is not a power of two, we obtain an approximate value of $\mathrm{Tr}_{\mathrm{RLLT}}(p)$ and $\mathrm{Tw}_{\mathrm{RLLT}}(p)$ by replacing 2^l with p and l with $\log_2(p)$:

$$\mathrm{Tr}_{\mathrm{RLLT}}(p) \approx \frac{p^3}{3} + \frac{p^2}{4} + p \left(\frac{3}{4}\log_2(p) + \frac{5}{12}\right),$$
$$\mathrm{Tw}_{\mathrm{RLLT}}(p) \approx \frac{3p^2}{4} + \frac{p}{4}\left(\log_2(p) + 1\right).$$

6.4. A one-tile implementation. A one-tile implementation of the recursive variant of Cholesky factorization is easily obtained by calling a one-tile implementation of OOC_TRSM instead of the two-tiles implementation analyzed in section 6.2.2. The number of tiles read by OOC_TRSM is slightly increased:

$$\mathrm{Tr}^{(1T)}_{\mathrm{TRSM}}(m,p) = m\,p^2 + m\,p,$$

whereas the number of tiles written is not modified. Finally, the number of tiles read by a one-tile recursive factorization of a matrix with 2^l tiles is

$$\mathrm{Tr}^{(1T)}_{\mathrm{RLLT}}\left(2^l\right) = \frac{2^{3l}}{3} + \frac{3}{4}\,2^{2l} + 2^l \left(\frac{l}{4} - \frac{1}{12}\right).$$

6.5. The hybrid recursive/left-looking approach. We investigate the hybrid implementation proposed in section 3.4.2. Assume that $p = 2^l$. This hybrid variant is very close to the fully recursive variant analyzed in section 6.3.

The recursive partitioning of the matrix is stopped at level $l-1$ (instead of l) when the matrix has 2×2 tiles.

The read operations corresponding to the level l are avoided. This results in 3×2^l tiles less to read. At level $l-1$, $3 \times 2^{l-1}$ additional tiles are read (for the factorization of the 2^{l-1} blocks of 2×2 tiles). Finally,

$$\mathrm{Tr}_{\mathrm{HLLT}}(p) = \mathrm{Tr}_{\mathrm{RLLT}}(p) - 3 \times 2^l + 3 \times 2^{l-1} = \frac{2^{3l}}{3} + \frac{2^{2l}}{4} + 2^l \left(\frac{3l}{4} - \frac{13}{12}\right).$$

This variant avoids to write 2^{l-1} tiles. Hence,

$$\mathrm{Tw}_{\mathrm{HLLT}}(p) = \mathrm{Tw}_{\mathrm{RLLT}}(p) - 2^{l-1} = \frac{3}{4}\,2^{2l} + 2^l \left(\frac{l-1}{4}\right).$$

The hybrid variant slightly improves the number of tiles read and written. Nevertheless, it does not affect the dominant terms.

TABLE 6.1
The number of tiles read and written during the factorization of an out-of-core matrix A partitioned into $p \times p$ tiles for the left-looking and the recursive variants of the Cholesky algorithm. For each variant, the one-tile and two-tiles implementation are considered.

	Left-looking algorithm	
Implementation	One-tile	Two-tiles
Tiles read	$\frac{p^3}{3} + \frac{p^2}{2} + \frac{p}{6}$	$\frac{p^3}{6} + \frac{2\,p}{3}$
Tiles written	$\frac{p}{2}(p+1)$	

	Recursive algorithm	
Implementation	One-tile	Two-tiles
Tiles read	$\frac{p^3}{3} + \frac{3}{4}p^2 + p\left(\frac{\log_2(p)}{4} - \frac{1}{12}\right)$	$\frac{p^3}{3} + \frac{p^2}{4} + p\left(\frac{3}{4}\log_2(p) + \frac{5}{12}\right)$
Tiles written	$\frac{3\,p^2}{4} + \frac{p}{4}(\log_2(p)+1)$	

6.6. Conclusion of the analysis. The number of tiles read from and written to the disk during the factorization of an out-of-core matrix A partitioned into $p \times p$ tiles is gathered in Table 6.1 for the left-looking and the recursive variants of the Cholesky algorithm. For each variant, the one-tile and two-tiles implementations are considered. The conclusion of this analysis is that the recursive variant reads and writes more tiles. Therefore, when one tries to minimize the number of tiles read and written, the left-looking variant is more appropriate.

7. Performance experiments. We present experiments for the different variants of the out-of-core Cholesky factorization. All variants were implemented in Fortran 90, compiled with Intel's Fortran compiler `ifort` with `-fast` optimization, and run on a Pentium 4 XEON based biprocessor running RedHat Linux 3.2.2-5. The 2 CPUs have a clock cycle of 3.05 GHz, 2GB of Ram, and a cache size of 512 KB. We use a small computer system interface (SCSI) disk for the out-of-core storage of matrices. We use Intel's optimized implementation of BLAS, the math kernel library. Experiments are performed in simple precision complex arithmetic with matrices generated by the discretization of boundary element formulation of the electromagnetic scattering by an object.

7.1. Narrow blocks. We present a preliminary study which aims at optimizing the width of narrow blocks.

We first verify the estimate (5.4) (estimate (5.5) is similarly verified). We consider the factorization of a matrix A of size $n = 12000$, arithmetically partitioned into tiles of size $t \times t$, by the two-tiles implementation of the Cholesky factorization. The memory is limited to 10^6 words. We compare the ratio of the number of terms read with two extreme choices:

- b minimum ($b = 1$), t maximum ($t = 707$): $\mathrm{Nr}_{\mathrm{LLT}}^{(2\mathrm{T})}(b = 1)$,
- b maximum, equal to t ($b = t = 500$): $\mathrm{Nr}_{\mathrm{LLT}}^{(2\mathrm{T})}(b = t = 500)$,

to the predicted value $1/\sqrt{2}$ and obtain a good agreement:

$$\frac{\mathrm{Nr}_{\mathrm{LLT}}^{(2\mathrm{T})}(b = 500)}{\mathrm{Nr}_{\mathrm{LLT}}^{(2\mathrm{T})}(b = 1)}\sqrt{2} = \frac{824\,10^6 \times \sqrt{2}}{1156\,10^6} = 1.0080.$$

But, when $b = 1$, the factorization takes 1858 seconds, against 398 seconds, when $b = t = 500$. When $b = 1$, the bulk of the computation is, namely, in level 2 BLAS, which explains this poor performance.

TABLE 7.1

I/O time (in seconds) for the factorization of a matrix of size $N = 48000$, partitioned into tiles of size $t \times t$, for different values of t. An "empty" factorization (with read/write operations but without computation) is performed. The elapsed time for the "empty" factorization roughly corresponds to the I/O time for the real factorization.

	Left-looking algorithm		Recursive algorithm		Hybrid algorithm	
t	One-tile	Two-tiles	One-tile	Two-tiles	One-tile	Two-tiles
750	4704	4951	5039	5125	5052	5122
1500	2727	2512	2900	2756	2841	2778
3000	1499	1543	1695	1713	1672	1625

We then follow [15] and choose b such that it optimizes an out-of-core SYRK. We compute $C - A\,A^T$, where matrices C and A consist of a single tile. We perform two sets of experiments: one with $t = 2535$, and the other with $t = 5880$. For each set, several values of b are tested (from $b = 128$ to $b = t/2$). The value $b = 256$ is a good compromise between speed and memory. This value is fixed throughout the following experiments.

7.2. Performance characteristics of the different variants. In this paragraph, we compare the performance characteristics of the left-looking, the recursive, and the hybrid variant of the Cholesky algorithm. The test case is a matrix of size $n = 48000$, partitioned in tiles of size 750, 1500, and 3000. For these tile sizes, arithmetic and recursive partitioning of the matrix are identical. One-tile and two-tiles implementations are experienced for each variant.

7.2.1. Evaluation of the time taken by the I/Os. The time taken by the I/Os is shown on Table 7.1. It is evaluated for each variant and each tile size by running an "empty" factorization of the matrix: the tiles are read and written according to the algorithmic schedule but no computation is made (all of the BLAS calls are turned off). The elapsed time gives an estimate of the time taken by the I/Os.

. . . 1. As mentioned in subsection 3.2, I/O operations are performed through standard read/write. But the operating system (OS) may cache pages in memory. Therefore, I/O timings may not be reliable. In this work, the specifier `BUFFERED='NO'` is used in the `OPEN` statements: this specifier is available in the Intel Fortran compiler, and it should ensure that records are not accumulated in a buffer instead of being read or written on disk. Nevertheless, the experimental timings presented here may be inaccurate, due to OS caching.

7.2.2. Performances of the different variants. Table 7.2 shows the performances of all of the variants. The elapsed time for completing the factorization is presented, along with the flops rate attained. The number of tiles read and written is measured, and this measure confirms the previous analysis of the I/O costs. The main result of these experiments is that the loop-based algorithm outperforms the others: it reads and writes less tiles, and it achieves the factorization faster than the other candidates. The recursive variant, although promising, does not seem to be well suited for an out-of-core decomposition. The I/O cost is higher than for the loop-based variant, and the better data locality provided by the recursive schedule does not compensate this drawback. The hybrid algorithm performs fewer I/O operations than the fully recursive one, but it does not perform better: the total time taken by the factorization is the same.

For a fixed given tile size, the two-tiles implementation is always faster than the one-tile implementation (for the three variants considered here). This is not a surprise:

TABLE 7.2

Factorization of a matrix of size $N = 48000$, partitioned into tiles of size $t \times t$, for different values of t. We compare one-tile and two-tiles implementations of the left-looking, recursive, and hybrid Cholesky factorization. For the hybrid algorithm, the recursive splitting of the matrix is stopped when the matrix has 2×2 tiles. Then, the algorithm switches in one case to the optimized (two-tiles) left-looking algorithm presented in section 3.4.2 and in the other case to the standard one-tile left-looking algorithm. We measure the number of tiles read and written during the factorization and the total time elapsed. Times are reported in seconds.

	Left-looking algorithm							
	One-tile implementation				Two-tiles implementation			
t	Tiles read	Tiles written	Gflops rate (Gflops/s)	Total time (s)	Tiles read	Tiles written	Gflops rate (Gflops/s)	Total time (s)
750	89440	2080	1.56	23590	87424	2080	1.55	23800
1500	11440	528	1.84	20025	10944	528	1.87	19723
3000	1496	136	1.99	18496	1376	136	2.08	17753
	Recursive algorithm							
	One-tile implementation				Two-tiles implementation			
t	Tiles read	Tiles written	Gflops rate (Gflops/s)	Total time (s)	Tiles read	Tiles written	Gflops rate (Gflops/s)	Total time (s)
750	90544	3184	1.54	23960	88720	3184	1.54	23991
1500	11728	816	1.83	20173	11312	816	1.85	19904
3000	1572	212	1.97	18679	1484	212	2.04	18070
	Hybrid algorithm							
	One-tile implementation				Two-tiles implementation			
t	Tiles read	Tiles written	Gflops rate (Gflops/s)	Total time (s)	Tiles read	Tiles written	Gflops rate (Gflops/s)	Total time (s)
750	90512	3152	1.54	23958	88624	3152	1.54	23900
1500	11712	800	1.82	20207	11264	800	1.85	19901
3000	1564	204	1.97	18650	1460	204	2.06	17929

a one-tile implementation performs better than a two-tiles implementation when the memory size (and not the tile size) is fixed.

8. Conclusion and perspectives. This paper analyses and compares out-of-core implementations of the partitioned Cholesky factorization algorithm. The same partitioning of the matrix to factorize is used for all variants, to allow a fair comparison. Our theoretical analysis shows that the amount of I/O operations is lower when a left-looking algorithm is used. Our numerical experiments confirm this result. Moreover, in our tests, the two-tiles implementation of the left-looking algorithm is the fastest variant, when tile size is fixed. It is the equivalent in complex arithmetic of the (sequential) POOCLAPACK routine for the Cholesky decomposition of real positive definite matrices [15]. Some improvements could still be added to this solver, such as prefetching the tiles, to overlap I/O operations and computation, as is performed in [22].

Acknowledgments. The author thanks the anonymous referees for their helpful comments and suggestions.

REFERENCES

[1] E. AGULLO, A. GUERMOUCHE, AND J.-Y. L'EXCELLENT, *A preliminary out-of-core extension of a parallel multifrontal solver*, in Euro-Par 2006 Parallel Processing, Springer-Verlag, New York, 2006, pp. 1053–1063.

[2] B. S. Andersen, J. A. Gunnels, F. G. Gustavson, J. K. Reid, and J. Wasniewski, *A fully portable high performance minimal storage hybrid format Cholesky algorithm*, ACM Trans. Math. Software, 31 (2005), pp. 201–227.

[3] B. S. Andersen, J. Wasniewski, and F. G. Gustavson, *A recursive formulation of Cholesky factorization of a matrix in packed storage*, ACM Trans. Math. Software, 27 (2001), pp. 214–244.

[4] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.

[5] M. Baboulin, L. Giraud, and S. Gratton, *A parallel distributed solver for large dense symmetric systems: Applications to geodesy and electromagnetism problems*, J. High Speed Comput., 19 (2005), pp. 353–363.

[6] A. Bendali, *Numerical analysis of the exterior boundary value problem for the time-harmonic Maxwell equations by a boundary finite element method, Part* 1: *The continuous problem, Part* 2: *The discrete problem*, Math. Comp., 43 (1984), pp. 29–68.

[7] N. Béreux, *A Cholesky Algorithm for Some Complex Symmetric Systems*, Technical report 515, CMAP, École Polytechnique, Palaiseau, France, 2003.

[8] S. H. Christiansen and J.-C. Nédélec, *A preconditioner for the electric field integral equation based on Calderon formulas*, SIAM J. Numer. Anal., 40 (2002), pp. 1100–1135.

[9] S. H. Christiansen, *Discrete Fredholm properties and convergence estimates for the electric field integral equation*, Math. Comp., 73 (2004), pp. 143–167.

[10] E. Darve, *The fast multipole method* (I): *Error analysis and asymptotic complexity*, SIAM J. Numer. Anal., 38 (2000), pp. 98–128.

[11] E. F. D'Azevedo and J. Dongarra, *The design and implementation of the parallel out-of-core ScaLAPACK LU, QR and Cholesky factorization routines*, Concurrency: Practice and Experience, 12 (2000), pp. 1481–1493.

[12] J. J. Dongarra, J. D. Croz, S. Hammarling, and I. Duff, *A set of level* 3 *basic linear algebra subprograms*, ACM Trans. Math. Software, 16 (1990), pp. 1–17.

[13] E. Elmroth, F. Gustavson, I. Jonsson, and B. Kågström, *Recursive blocked algorithms and hybrid data structures for dense matrix library software*, SIAM Rev., 46 (2004), pp. 3–45.

[14] J. A. Gunnels and R. A. van de Geijn, *Developing Linear Algebra Algorithms: A Collection of Class Projects*, FLAME Working Note 3, Department of Computer Sciences, The University of Texas, Austin, TX, 2001.

[15] B. C. Gunter, W. C. Reiley, and R. A. van de Geijn, *Parallel Out-of-Core Cholesky and QR Factorization with POOCLAPACK*, in Proceedings of the 15th IPDPS, IEEE Computer Society, 2001, p. 179.

[16] F. G. Gustavson and I. Jonsson, *Minimal storage high performance Cholesky factorization via blocking and recursion*, IBM J. Res. Develop., 44 (2000), pp. 823–849.

[17] D. Irony, G. Shklarski, and S. Toledo, *Parallel and fully recursive multifrontal sparse Cholesky*, Future Generation Comput. Syst., 20 (2004), pp. 425–440.

[18] T. Joffrain, E. S. Quintana-Ortí, and R. A. van de Geijn, *Rapid development of high-performance out-of-core solvers*, in PARA 2004, Lyngby, Denmark, 2004, pp. 413–422.

[19] J.-C. Nédélec, *Acoustic and electromagnetic equations: Integral representations for harmonic problems*, Appl. Math. Sci. 144, Springer-Verlag, New York, 2001.

[20] E. S. Quintana, G. Quintana, X. Sun, and R. van de Geijn, *A note on parallel matrix inversion*, SIAM J. Sci. Comput., 22 (2001), pp. 1762–1771.

[21] W. C. Reiley and R. A. van de Geijn, *POOCLAPACK: Parallel out-of-core linear algebra package*, Technical report CS-TR-99-33, Department of Computer Sciences, The University of Texas at Austin, Austin, TX, 1999.

[22] S. Toledo and F. G. Gustavson, *The design and implementation of SOLAR, a portable library for scalable out-of-core linear algebra computations*, in IOPADS '96: Proceedings of the Fourth Workshop on I/O in Parallel and Distributed Systems, ACM Press, New York, 1996, pp. 28–40.

[23] S. Toledo, *Locality of reference in LU decomposition with partial pivoting*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1065–1081.

[24] S. Toledo, *A survey of out-of-core algorithms in numerical linear algebra*, in External Memory Algorithms and Visualization, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., J. Abello and J. S. Vitter, eds., American Mathematical Society Press, Providence, RI, 1999, pp. 161–180.

[25] R. A. van de Geijn, *Using PLAPACK*, The MIT Press, Cambridge, MA, 1997.

[26] R. Wesley, *Efficient Parallel Out-of-core Implementation of the Cholesky Factorization*, PLAPACK Working Note 11, Department of Computer Sciences, The University of Texas at Austin, Austin, TX, December 1999.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1320–1338

OPTIMAL SCALING OF GENERALIZED AND POLYNOMIAL EIGENVALUE PROBLEMS*

T. BETCKE†

Abstract. Scaling is a commonly used technique for standard eigenvalue problems to improve the sensitivity of the eigenvalues. In this paper we investigate scaling for generalized and polynomial eigenvalue problems (PEPs) of arbitrary degree. It is shown that an optimal diagonal scaling of a PEP with respect to an eigenvalue can be described by the ratio of its normwise and componentwise condition number. Furthermore, the effect of linearization on optimally scaled polynomials is investigated. We introduce a generalization of the diagonal scaling by Lemonnier and Van Dooren to PEPs that is especially effective if some information about the magnitude of the wanted eigenvalues is available and also discuss variable transformations of the type $\lambda = \alpha\mu$ for PEPs of arbitrary degree.

Key words. polynomial eigenvalue problem, balancing, scaling, condition number, backward error

AMS subject classifications. 65F15, 15A18

DOI. 10.1137/070704769

1. Introduction.

Scaling of standard eigenvalue problems $Ax = \lambda x$ is a well-established technique that is implemented in the LAPACK routine `xGEBAL`. It goes back to work by Osborne [14] and Parlett and Reinsch [15]. The idea is to find a diagonal matrix D that scales the rows and columns of $A \in \mathbb{C}^{n\times n}$ in a given norm such that

$$\|D^{-1}ADe_i\| = \|e_i^*D^{-1}AD\|, \quad i = 1, \ldots, n,$$

where e_i is the ith unit vector. This is known as balancing. LAPACK uses the 1-norm. Balancing matrix rows and columns can often reduce the effect of rounding errors on the computed eigenvalues. However, as Watkins demonstrated [19], there are also cases in which balancing can lead to a catastrophic increase of the errors in the computed eigenvalues.

For generalized eigenvalue problems (GEPs) $Ax = \lambda Bx$, a scaling technique proposed by Ward [18] is implemented in the LAPACK routine `xGGBAL`. Its aim is to find diagonal matrices D_1 and D_2 such that the elements of D_1AD_2 and D_1BD_2 are scaled as equal in magnitude as possible.

A different approach for the scaling of GEPs is proposed by Lemonnier and Van Dooren [11]. In section 5 we will come back to this. It is interesting to note that the default behavior of LAPACK (and also of MATLAB) is to scale nonsymmetric standard eigenvalue problems but not to scale GEPs.

In this paper we discuss the scaling of polynomial eigenvalue problems (PEPs) of the form

$$P(\lambda)x := (\lambda^\ell A_\ell + \cdots + \lambda A_1 + A_0)x = 0, \quad A_k \in \mathbb{C}^{n\times n},\ A_\ell \neq 0,\ \ell \geq 1. \tag{1.1}$$

Every $\lambda \in \mathbb{C}$ for which there exists a solution $x \in \mathbb{C}^n\backslash\{0\}$ of $P(\lambda)x = 0$ is called an

*Received by the editors October 8, 2007; accepted for publication (in revised form) by L. De Lathauwer April 30, 2008; published electronically October 16, 2008. This work was supported by Engineering and Physical Sciences Research Council grant EP/D079403/1.
http://www.siam.org/journals/simax/30-4/70476.html
†School of Mathematics, The University of Manchester, Oxford Road, Manchester, M13 9PL, UK (timo.betcke@maths.man.ac.uk, http://www.ma.man.ac.uk/~tbetcke).

eigenvalue of P with associated right eigenvector x. We will also need left eigenvectors $y \in \mathbb{C}^n \backslash \{0\}$ defined by $y^* P(\lambda) = 0$.

In section 2 we review the definition of condition numbers and backward errors for the PEP (1.1). Then in section 3 we investigate diagonal scalings of (1.1) of the form $D_1 P(\lambda) D_2$, where D_1 and D_2 are diagonal matrices in the set

$$\mathcal{D}_n := \{D : \ D \in \mathbb{C}^{n \times n} \text{ is diagonal and } \det(D) \neq 0\}.$$

We show that the minimal achievable normwise condition number of an eigenvalue by diagonal scaling of $P(\lambda)$ can be bounded by its componentwise condition number. This gives easily computable conditions on whether the condition number of eigenvalues can be improved by scaling. The results of that section can be applied to generalized linear and higher degree polynomial problems.

The most widely used technique to solve PEPs of degree $\ell \geq 2$ is to convert the associated matrix polynomial into a linear pencil, the process of linearization, and then solve the corresponding GEP. In section 4 we investigate the difference between scaling before or after linearizing the matrix polynomial. Then in section 5 we introduce a heuristic scaling strategy for PEPs that generalizes the idea of Lemonnier and Van Dooren. It is applicable to arbitrary polynomials of degree $\ell \geq 1$ and includes a weighting factor that, given some information about the magnitude of the wanted eigenvalues, can crucially improve the normwise condition numbers of eigenvalues after scaling.

Fan, Lin and Van Dooren [3] propose a transformation of variables of the form $\lambda = \alpha\mu$ for some parameter α for quadratic polynomials whose aim is to improve the backward stability of numerical methods for quadratic eigenvalue problems (QEPs) that are based on linearization. In section 6 we extend this variable transformation to matrix polynomials of arbitrary degree $\ell \geq 2$.

Numerical examples illustrating our scaling algorithms are presented in section 7. We conclude with practical remarks on how to put the results of this paper into practice.

Scaling routines for standard and generalized eigenvalue problems often include a preprocessing step that attempts to remove isolated eigenvalues by permutation of the matrices. This is, for example, implemented in the LAPACK routines `xGBAL` and `xGGBAL`. Since the permutation algorithm described in [18] can be easily adapted for matrix polynomials, we will not discuss this further in this paper. But nevertheless, it is advisable to use this preprocessing step also for PEPs to reduce the problem dimension if possible.

All notation is standard. For a matrix A, we denote by $|A|$ the matrix of absolute values of the entries of A. Similarly, $|x|$ for a vector x denotes the absolute values of the entries of x. The vector of all ones is denoted by e; that is, $e = \begin{bmatrix} 1, 1, \ldots, 1 \end{bmatrix}^T \in \mathbb{R}^n$.

2. Normwise and componentwise error bounds. An important tool to measure the quality of an approximate eigenpair $(\tilde{x}, \tilde{\lambda})$ of the PEP $P(\lambda)x = 0$ is its normwise backward error. With $\Delta P(\lambda) = \sum_{k=0}^{\ell} \lambda^k \Delta A_k$ it is defined for the 2-norm by

$$\eta_P\left(\tilde{x}, \tilde{\lambda}\right) := \min\left\{\epsilon : \ \left(P\left(\tilde{\lambda}\right) + \Delta P\left(\tilde{\lambda}\right)\right)\tilde{x} = 0, \ \|\Delta A_k\|_2 \leq \epsilon \|A_k\|_2, \ k = 0 : \ell\right\}.$$

Tisseur [17] shows that

$$\eta_P\left(\tilde{x}, \tilde{\lambda}\right) = \frac{\|r\|_2}{\tilde{\alpha}\|\tilde{x}\|_2},$$

where $r = P(\tilde{\lambda})\tilde{x}$ and $\tilde{\alpha} = \sum_{k=0}^{\ell} |\tilde{\lambda}|^k \|A_k\|_2$. The normwise backward error $\eta(\tilde{\lambda})$ of a computed eigenvalue $\tilde{\lambda}$ is defined as

$$\eta_P\left(\tilde{\lambda}\right) = \min_{\substack{x \in \mathbb{C}^n \\ x \neq 0}} \eta_P\left(x, \tilde{\lambda}\right).$$

It follows immediately [17, Lemma 3] that $\eta_P(\tilde{\lambda}) = (\tilde{\alpha}\|P(\tilde{\lambda})^{-1}\|_2)^{-1}$.

The sensitivity of an eigenvalue is measured by the condition number. It relates the forward error, that is, the error in the computed eigenvalue $\tilde{\lambda}$, and the backward error $\eta_P(\tilde{\lambda})$. To first order (meaning up to higher terms in the backward error) one has

$$\text{forward error} \leq \text{backward error} \times \text{condition number}. \tag{2.1}$$

The condition number of a simple, finite, nonzero eigenvalue $\lambda \neq 0$ is defined by

$$\kappa_P(\lambda) := \lim_{\epsilon \to 0} \sup \left\{ \frac{|\Delta\lambda|}{\epsilon|\lambda|} \ : \ \big(P(\lambda+\Delta\lambda) + \Delta P(\lambda+\Delta\lambda)\big)(x+\Delta x) = 0, \right.$$
$$\left. \|\Delta A_k\|_2 \leq \epsilon \|A_k\|_2, \ k = 0:\ell \right\}.$$

Let x be a right eigenvector and y be a left eigenvector associated with the eigenvalue λ of P. Then $\kappa_P(\lambda)$ is given by [17, Theorem 5]

$$\kappa_P(\lambda) = \frac{\|y\|_2 \|x\|_2 \alpha}{|y^* P'(\lambda) x||\lambda|}, \quad \alpha = \sum_{k=0}^{\ell} |\lambda|^k \|A_k\|_2. \tag{2.2}$$

Backward error and condition number can also be defined in a componentwise sense. The componentwise backward error of an eigenpair $(\tilde{x}, \tilde{\lambda})$ is

$$\omega_P\left(\tilde{x}, \tilde{\lambda}\right) := \min\left\{\epsilon : \ \left(P\left(\tilde{\lambda}\right) + \Delta P\left(\tilde{\lambda}\right)\right)\tilde{x} = 0; |\Delta A_k| \leq \epsilon |A_k|, \ k = 0:\ell\right\}. \tag{2.3}$$

The componentwise condition number of a simple, finite, nonzero eigenvalue λ is defined as

$$\operatorname{cond}_P(\lambda) := \lim_{\epsilon \to 0} \sup \left\{ \frac{|\Delta\lambda|}{\epsilon|\lambda|} \ : \ \big(P(\lambda+\Delta\lambda) + \Delta P(\lambda+\Delta\lambda)\big)(x+\Delta x) = 0, \right.$$
$$\left. |\Delta A_k| \leq \epsilon |A_k|, \ k = 0:\ell \right\}. \tag{2.4}$$

The following theorem gives explicit expressions for these quantitites.

THEOREM 2.1. [illegible] $(\tilde{x}, \tilde{\lambda})$ [illegible]

$$\omega_P(\tilde{x}, \tilde{\lambda}) = \max_i \frac{|r_i|}{\left(\widetilde{A}|\tilde{x}|\right)_i}, \quad \widetilde{A} := \sum_{k=0}^{\ell} |\tilde{\lambda}|^k |A_k|, \tag{2.5}$$

r_i ith $P(\tilde{\lambda})\tilde{x}$ λ y x

$$\text{cond}_P(\lambda) = \frac{|y|^*A|x|}{|\lambda||y^*P'(\lambda)x|}, \quad A := \sum_{k=0}^{\ell} |\lambda|^k |A_k|. \tag{2.6}$$

. The proof is a slight modification of the proofs of [5, Theorems 3.1 and 3.2] along the lines of the proof of [17, Theorem 1]. ∎

Surveys of componentwise error analysis are contained in [8, 9]. The componentwise backward error and componentwise condition number are invariant under multiplication of $P(\lambda)$ from the left and the right with nonsingular diagonal matrices. In the next section we will use this property to characterize optimally scaled eigenvalue problems.

3. Optimal scalings. In this section we introduce the notion of an optimal scaling with respect to a certain eigenvalue and give characterizations of it.

Ultimately, we are interested in computing eigenvalues to as many digits as possible. Hence, we would like to find a scaling that leads to small forward errors. If we assume that we use a backward stable algorithm, that is, the backward error is only a small multiple of the machine precision, then it follows from (2.1) that we can hope to compute an eigenvalue to many digits of accuracy by finding a scaling that minimizes the condition number.

In the following we define what we mean by a scaling of a matrix polynomial $P(\lambda)$.

DEFINITION 3.1. $P(\lambda) \in \mathbb{C}^{n\times n}$ $P(\lambda)$ $D_1P(\lambda)D_2$ $D_1, D_2 \in \mathcal{D}_n$

It is immediately clear that the eigenvalues of a matrix polynomial $P(\lambda)$ are invariant under scaling. Furthermore, if (y, x, λ) is an eigentriplet of $P(\lambda)$ with eigenvalue λ and left and right eigenvector y and x, respectively, then an eigentriplet of the scaling $D_1P(\lambda)D_2$ is $(D_1^{-*}y, D_2^{-1}x, \lambda)$.

The following definition defines an optimal scaling of $P(\lambda)$ with respect to a given eigenvalue λ in terms of minimizing the condition number of λ.

DEFINITION 3.2. λ $P(\lambda)$ $P(\lambda)$ λ

$$\kappa_P(\lambda) = \inf_{D_1, D_2 \in \mathcal{D}_n} \kappa_{D_1PD_2}(\lambda).$$

This definition of optimal scaling depends on the eigenvalue λ. We cannot expect that an optimal scaling for one eigenvalue also gives an optimal scaling for another eigenvalue. The following theorem states that a PEP is almost optimally scaled with respect to an eigenvalue λ, if the componentwise and normwise condition numbers of λ are close to each other. Furthermore, it gives explicit expressions for scaling matrices $D_1, D_2 \in \mathcal{D}_n$ that achieve an almost optimal scaling.

THEOREM 3.3. λ $n \times n$ $P(\lambda)$ y x

$$\frac{1}{\sqrt{n}}\,\text{cond}_P(\lambda) \le \inf_{D_1, D_2 \in \mathcal{D}_n} \kappa_{D_1PD_2}(\lambda) \le n\,\text{cond}_P(\lambda). \tag{3.1}$$

y x

$$D_1 = \text{diag}(|y|), \quad D_2 = \text{diag}(|x|),$$

$$\kappa_{D_1PD_2}(\lambda) \leq n \operatorname{cond}_P(\lambda). \tag{3.2}$$

. Let $A := \sum_{k=0}^{\ell} |\lambda|^k |A_k|$ and $\alpha := \sum_{k=0}^{\ell} |\lambda|^k \|A_k\|_2$. Using $\||B|\|_2 \leq \sqrt{n}\|B\|_2$ [9, Lemma 6.6] for any matrix $B \in \mathbb{C}^{n\times n}$, the lower bound follows from

$$\operatorname{cond}_P(\lambda) = \frac{|y|^*A|x|}{|\lambda||y^*P'(\lambda)x|} \leq \frac{\|y\|_2\|x\|_2\|A\|_2}{|\lambda||y^*P'(\lambda)x|} \leq \frac{\sqrt{n}\alpha\|y\|_2\|x\|_2}{|\lambda||y^*P'(\lambda)x|} = \sqrt{n}\kappa_P(\lambda),$$

and the fact that the componentwise condition number is invariant under diagonal scaling. For $\epsilon > 0$ define the vectors $\tilde{y}$ and $\tilde{x}$ by

$$\tilde{y}_i = \begin{cases} y_i, & y_i \neq 0 \\ \epsilon, & y_i = 0 \end{cases} \qquad \tilde{x}_i = \begin{cases} x_i, & x_i \neq 0 \\ \epsilon, & x_i = 0 \end{cases}$$

and consider the diagonal matrices

$$D_1 = \operatorname{diag}(|\tilde{y}|), \quad D_2 = \operatorname{diag}(|\tilde{x}|).$$

Using $\|B\|_2 \leq e^*|B|e$ for any matrix $B \in \mathbb{C}^{n\times n}$ [9, Table 6.2], we have

$$\kappa_{D_1PD_2}(\lambda) = \frac{\|D_1^{-1}y\|_2\|D_2^{-1}x\|_2 \left(\sum_{k=0}^{\ell} |\lambda|^k \|D_1A_kD_2\|_2\right)}{|\lambda||y^*P'(\lambda)x|} \leq \frac{n\left(\sum_{k=0}^{\ell} |\lambda|^k e^*|D_1A_kD_2|e\right)}{|\lambda||y^*P'(\lambda)x|}$$

$$= \frac{n\left(\sum_{k=0}^{\ell} |\lambda|^k \cdot |\tilde{y}|^* \cdot |A_k| \cdot |\tilde{x}|\right)}{|\lambda||y^*P'(\lambda)x|} \longrightarrow n \operatorname{cond}_P(\lambda) \text{ as } \epsilon \to 0. \tag{3.3}$$

The upper bounds in (3.1) and (3.2) follow immediately. ∎

Theorem 3.3 is restricted to finite and nonzero eigenvalues. Assume that $\lambda = 0$ is an eigenvalue. Then we have to replace relative componentwise and normwise condition numbers by the absolute condition numbers

$$\kappa_P^{(a)}(\lambda) = \frac{\|y\|_2\|x\|_2\alpha}{|y^*P'(\lambda)x|}, \quad \operatorname{cond}_P^{(a)}(\lambda) = \frac{|y|^*A|x|}{|y^*P'(\lambda)x|}.$$

With these condition numbers Theorem 3.3 is also valid for zero eigenvalues. If $P(\lambda)$ has an infinite eigenvalue, the reversal $\operatorname{rev} P(\lambda) := \lambda^\ell P(1/\lambda)$ has a zero eigenvalue, and we can apply Theorem 3.3 using absolute condition numbers to $\operatorname{rev} P(\lambda)$.

While Theorem 3.3 applies to generalized linear and polynomial problems, it does not immediately apply to standard problems of the form $Ax = \lambda x$. The crucial difference is that for standard eigenvalue problems we assume the right-hand side identity matrix to be fixed and only allow scalings of the form $D^{-1}AD$ that leave the identity unchanged. However, if λ is an eigenvalue of A with associated left and right eigenvectors y and x that have nonzero entries, we can still define $D_1 = \operatorname{diag}(|y|)$ and $D_2 = \operatorname{diag}(|x|)$ to obtain the generalized eigenvalue problem

$$D_1AD_2v = \lambda D_1D_2v, \tag{3.4}$$

where $x = D_2v$. Since D_1D_2 has positive diagonal entries there exists $\hat{D}$ such that $\hat{D}^2 = D_1D_2$. We then obtain from (3.4) the standard eigenvalue problem

$$\hat{D}^{-1}D_1AD_2\hat{D}^{-1}\tilde{x} = \lambda\tilde{x}, \tag{3.5}$$

where $\tilde{x} = \hat{D}D_2^{-1}x$ and $|\tilde{x}| = \left[\sqrt{|y_1||x_1|}, \ldots, \sqrt{|y_n||x_n|}\right]^T$. For the scaled left eigenvector $\tilde{y}$ we have $\tilde{y} = \hat{D}D_1^{-1}y$ and $|\tilde{y}| = |\tilde{x}|$. If we define the normwise condition number $k_A(\lambda)$ and the componentwise condition number $c_A(\lambda)$ for the eigenvalue λ of a standard eigenvalue problem by[1]

$$k_A(\lambda) = \frac{\|y\|_2\|x\|_2}{|\lambda||y^Tx|}, \quad c_A(\lambda) = \frac{|y|^T|x|}{|\lambda||y^Tx|},$$

it follows for the scaling $D^{-1}AD$, where $D = D_2\hat{D}^{-1} = D_1^{-1}\hat{D}$ that

$$c_A(\lambda) = k_{D^{-1}AD}(\lambda).$$

But this scaling is not always useful as $\|D^{-1}AD\|_2$ can become large if x or y contain tiny entries.

There is a special case in which the scaling (3.5) also minimizes $\|D^{-1}AD\|_2$. If λ is the Perron root of an irreducible and nonnegative matrix A, the corresponding left and right eigenvectors y and x can be chosen to have positive entries. After scaling by D as described above, we have $k_{D^{-1}AD}(\lambda) = 1$ and $\|D^{-1}AD\|_2 = \lambda$. This was investigated by Chen and Demmel in [2] who proposed a weighted balancing which is identical to the scaling described above for nonnegative and irreducible A.

Theorem 3.3 gives us an easy way to check whether a matrix polynomial P is nearly optimally scaled with respect to an eigenvalue λ. We only need to compute the ratio

$$\frac{\kappa_P(\lambda)}{\operatorname{cond}_P(\lambda)} = \frac{\|y\|_2\|x\|_2 \sum_{k=0}^{\ell} |\lambda|^k \|A_k\|_2}{|y|^* \left(\sum_{k=0}^{\ell} |\lambda|^k |A_k|\right) |x|} \tag{3.6}$$

after computing the eigenvalues and eigenvectors. If an eigensolver already returns normwise condition numbers, this is only a little extra effort. If $\frac{\kappa_P(\lambda)}{\operatorname{cond}_P(\lambda)} \gg n$ the eigensolver can give a warning to the user that the problem is badly scaled and that the error in the computed eigenvalue λ is likely to become smaller by rescaling P. Furthermore, from Theorem 3.3 it follows that a polynomial is nearly optimally scaled if the entries of the left and right eigenvectors have equal magnitude. This motivates a heuristic scaling algorithm, which is discussed in section 5.

At the end we would like to emphasize that a diagonal scaling which improves the condition numbers of the eigenvalues needs not necessarily be a good scaling for eigenvectors. An example is the generalized linear eigenvalue problem $L(\lambda)x = 0$, where

$$L(\lambda) = \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 1 + 2 \cdot 10^{-8} & 2 \\ 2 & 10^{-8} & 1 \\ 1 & 1 + 10^{-8} & -1 \end{bmatrix}.$$

One eigenvalue is $\lambda = 1$ with associated right eigenvector $x = \begin{bmatrix} 1 & -1 & 10^{-8} \end{bmatrix}^T$ and left eigenvector $y = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & -1 \end{bmatrix}^T$. The condition number[2] of the eigenvector x before scaling is approximately 33.1. After scaling with $D_1 = \operatorname{diag}(|y|)$ and $D_2 = \operatorname{diag}(|x|)$, it increases to $1.70 \cdot 10^9$. For the corresponding eigenvalue $\lambda = 1$ we have $\kappa_L(1) \approx 21.8$ and after scaling $\kappa_{D_1LD_2}(1) \approx 19.6$. However, in most of our experiments we could not observe an increase of the eigenvector condition number after scaling.

[1] Choose $E = I$, $F = 0$, and $B = I$ in Theorems 2.5 and 3.2 of [5].

[2] The condition number of the eigenvector was computed using Theorem 2.7 from [5] with the normalization vector $g = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$.

4. Scalings and linearizations. The standard way to solve the PEP (1.1) of degree $\ell \geq 2$ is to convert $P(\lambda)$ into a linear pencil

$$L(\lambda) = \lambda X + Y$$

having the same spectrum as $P(\lambda)$ and then solve the eigenproblem for L. Formally, $L(\lambda)$ is a linearization if

$$E(\lambda)L(\lambda)F(\lambda) = \begin{bmatrix} P(\lambda) & 0 \\ 0 & I_{(\ell-1)n} \end{bmatrix}$$

for some unimodular $E(\lambda)$ and $F(\lambda)$ [4, section 7.2]. For example,

$$C_1(\lambda) = \lambda \begin{bmatrix} A_\ell & 0 & \cdots & 0 \\ 0 & I_n & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & I_n \end{bmatrix} + \begin{bmatrix} A_{\ell-1} & A_{\ell-2} & \cdots & A_0 \\ -I_n & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & -I_n & 0 \end{bmatrix} \tag{4.1}$$

is a linearization of $P(\lambda)$, called the first companion form. In [12] Mackey, Mackey, Mehl, and Mehrmann identified two vector spaces of pencils that are potential linearizations of $P(\lambda)$. Let $\Lambda := [\lambda^{\ell-1}, \lambda^{\ell-2}, \ldots, 1]^T$. Then these spaces are defined by

$$\begin{aligned} \mathbb{L}_1(P) &= \left\{ L(\lambda) : L(\lambda)(\Lambda \otimes I_n) = v \otimes P(\lambda),\ v \in \mathbb{C}^\ell \right\}, \\ \mathbb{L}_2(P) &= \left\{ L(\lambda) : (\Lambda^T \otimes I_n)L(\lambda) = \tilde{v}^T \otimes P(\lambda),\ \tilde{v} \in \mathbb{C}^\ell \right\}. \end{aligned}$$

The first companion linearization belongs to $\mathbb{L}_1(P)$ with $v = e_1$. Furthermore, the pencils in $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$ that are not linearizations form a closed nowhere dense subset of measure zero in these spaces [12, Theorem 4.7].

Another important space of potential linearizations is given by

$$\mathbb{DL}(P) := \mathbb{L}_1(P) \cap \mathbb{L}_2(P).$$

In [12, Theorem 5.3] it is shown that each pencil $L(\lambda) \in \mathbb{DL}(P)$ is uniquely defined by a vector $v \in \mathbb{C}^\ell$ such that

$$L(\lambda)(\Lambda \otimes I_n) = v \otimes P(\lambda), \quad \left(\Lambda^T \otimes I_n\right) L(\lambda) = v^T \otimes P(\lambda).$$

There is a well-defined relationship between the eigenvectors of linearizations $L(\lambda) \in \mathbb{DL}(P)$ and eigenvectors of $P(\lambda)$; namely, for finite eigenvalues λ x is a right eigenvector of $P(\lambda)$ if and only if $\Lambda \otimes x$ is a right eigenvector of $L(\lambda)$ and y is a left eigenvector of $P(\lambda)$ if and only if $\overline{\Lambda} \otimes y$ is a left eigenvector of $L(\lambda)$ [12, Theorems 3.8 and 3.14].

A simple observation is that scaling $P(\lambda)$ leads to a scaling of $L(\lambda)$ within the same space of potential linearizations.

LEMMA 4.1. *If $L(\lambda) \in \mathbb{S}(P)$ for some v, then $\mathbb{S}(P) = \mathbb{L}_1(P)$, $\mathbb{L}_2(P)$, $\mathbb{DL}(P)$ and $(I_n \otimes D_1)L(\lambda)(I_n \otimes D_2) \in \mathbb{S}(D_1 P D_2)$ for some v, $D_1, D_2 \in \mathbb{C}^{n \times n}$.*

Proof. The statements follow immediately from the identities

$$(I \otimes D_1)L(\lambda)(I \otimes D_2)(\Lambda \otimes I_n) = v \otimes D_1 P(\lambda) D_2$$

and

$$\left(\Lambda^T \otimes I_n\right)(I \otimes D_1)L(\lambda)(I \otimes D_2) = \tilde{v}^T \otimes D_1 P(\lambda) D_2$$

for matrices $D_1, D_2 \in \mathbb{C}^{n\times n}$. ∎

Hence, if we solve a PEP by a linearization in $\mathbb{L}_1(P)$, $\mathbb{L}_2(P)$, or $\mathbb{DL}(P)$, scaling of the original polynomial $P(\lambda)$ with matrices D_1 and D_2 is just a special scaling of the linearization $L(\lambda)$ with scaling matrices $(I \otimes D_1)$ and $(I \otimes D_2)$. If preserving structure of the linearization is not an issue, we can scale the linearization $L(\lambda)$ directly with diagonal scaling matrices $\widetilde{D}_1$ and $\widetilde{D}_2$ that have $2\ell n$ free parameters compared to the $2n$ free parameters in D_1 and D_2. The following theorem gives a worst case bound on the ratio between the optimal condition numbers with the two different scaling strategies (i.e., scaling and then linearizing or linearizing and then scaling).

THEOREM 4.2. [illegible] P [illegible] $L(\lambda) \in \mathbb{DL}(P)$ [illegible] v [illegible]

$$\inf_{D_1, D_2 \in \mathcal{D}_n} \kappa_{\widetilde{L}}(\lambda; v; D_1 P D_2) \le \begin{cases} \ell^{1/2} n^{3/2} \left(\frac{|\lambda|^{2\ell}-1}{|\lambda|^2-1}\right) \inf\limits_{\tilde{D}_1, \tilde{D}_2 \in \mathcal{D}_{\ell n}} \kappa_{\tilde{D}_1 L \tilde{D}_2}(\lambda) & \text{if } |\lambda| \ge 1, \\ \frac{\ell^{1/2} n^{3/2}}{|\lambda|^{2(\ell-1)}} \left(\frac{|\lambda|^{2\ell}-1}{|\lambda|^2-1}\right) \inf\limits_{\tilde{D}_1, \tilde{D}_2 \in \mathcal{D}_{\ell n}} \kappa_{\tilde{D}_1 L \tilde{D}_2}(\lambda) & \text{if } |\lambda| < 1, \end{cases}$$

[illegible] $\kappa_{\widetilde{L}}(\lambda; v; D_1 P D_2)$ [illegible] λ [illegible] $\widetilde{L}(\lambda) \in \mathbb{DL}(D_1 P D_2)$ [illegible] v

[illegible]. Let y and x be left and right eigenvectors of $P(\lambda)$ associated with the eigenvalue λ. Since $L(\lambda) = \lambda X + Y \in \mathbb{DL}(P)$, its left and right eigenvectors associated with λ are $\overline{\Lambda} \otimes y$ and $\Lambda \otimes x$. Assume that y and x have no zero entries. The case of zero entries follows by a limit process similar to that in the proof of Theorem 3.3. Define $D_1 = \operatorname{diag}(|y|)$ and $D_2 = \operatorname{diag}(|x|)$. Since $\|\overline{\Lambda} \otimes (D_1^{-1} y)\|_2 = \|\Lambda\|_2 \|D_1^{-1} y\|_2$ and $\|\Lambda \otimes (D_2^{-1} x)\|_2 = \|\Lambda\|_2 \|D_2^{-1} x\|_2$, we have

$$\begin{aligned} &\kappa_{\widetilde{L}}(\lambda; v; D_1 P D_2) \\ &= \frac{\|\Lambda\|_2^2 \|D_1^{-1} y\|_2 \|D_2^{-1} x\|_2 (|\lambda| \|(I \otimes D_1) X (I \otimes D_2)\|_2 + \|(I \otimes D_1) Y (I \otimes D_2)\|_2)}{|\lambda| \left|\left(\overline{\Lambda} \otimes y\right)^* X (\Lambda \otimes x)\right|}, \end{aligned}$$

and therefore by using $\|B\|_2 \le e^* |B| e$ for any $B \in \mathbb{C}^{n\times n}$

$$\kappa_{\widetilde{L}}(\lambda; v; D_1 P D_2) \le \frac{\|\Lambda\|_2^2 n \hat{e}^* (|\lambda| |(I \otimes D_1) X (I \otimes D_2)| + |(I \otimes D_1) Y (I \otimes D_2)|) \hat{e}}{|\lambda| \left|\left(\overline{\Lambda} \otimes y\right)^* X (\Lambda \otimes x)\right|}$$

for $\hat{e} = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T \in \mathbb{R}^{\ell n}$. Assume that $|\lambda| \ge 1$. Since componentwise $\hat{e} \le |\Lambda| \otimes e = |\overline{\Lambda}| \otimes e$ and

$$\left(|\overline{\Lambda}| \otimes e\right)^* (I \otimes D_1) = |\overline{\Lambda} \otimes y|^*, \quad (I \otimes D_2)(|\Lambda| \otimes e) = |\Lambda \otimes x|,$$

we obtain

$$\kappa_{\widetilde{L}}(\lambda; v; D_1 P D_2) \le n \frac{\|\Lambda\|_2^2 |\overline{\Lambda} \otimes y|^* (|\lambda| |X| + |Y|) |\Lambda \otimes x|}{|\lambda| \left|\left(\overline{\Lambda} \otimes y\right)^* X (\Lambda \otimes x)\right|} = n \|\Lambda\|_2^2 \operatorname{cond}_L(\lambda). \tag{4.2}$$

It holds that

$$\|\Lambda\|_2^2 = \left(\frac{|\lambda|^{2\ell} - 1}{|\lambda|^2 - 1}\right). \tag{4.3}$$

Furthermore, from Theorem 3.3 we know that

$$\frac{1}{\sqrt{\ell n}}\mathrm{cond}_L(\lambda) \leq \inf_{\tilde{D}_1,\tilde{D}_2\in\mathcal{D}_{\ell n}} \kappa_{\tilde{D}_1 L \tilde{D}_2}(\lambda). \tag{4.4}$$

Combining (4.2), (4.3), and (4.4), the proof for the case $|\lambda| \geq 1$ follows. The proof for $|\lambda| < 1$ is similar. The only essential difference is that now componentwise $\hat{e} \leq \frac{|\Lambda|}{|\lambda|^{\ell-1}} \otimes e$. ∎

Theorem 4.2 suggests that for eigenvalues that are large or small in magnitude first linearizing and then scaling can in the worst case be significantly better than first scaling and then linearizing. However, if we first linearize and then scale, the special structure of the linearization is lost.

How sharp are these bounds? In the following we discuss the case $|\lambda| \geq 1$. For the case $|\lambda| < 1$ analogous arguments can be used. Consider the QEP $Q(\lambda) = \lambda^2 A + \lambda B + C$, where

$$A = \begin{bmatrix} -0.6 & -0.1 \\ 2 & 0.1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & -0.1 \\ 0.6 & -0.8 \end{bmatrix}, \quad C = \begin{bmatrix} 3\cdot 10^7 & 7\cdot 10^7 \\ -1\cdot 10^8 & 1.6\cdot 10^8 \end{bmatrix}, \tag{4.5}$$

and its linearization in $\mathbb{DL}(Q)$

$$L(\lambda) = \lambda X + Y := \lambda \begin{bmatrix} A & 0 \\ 0 & -C \end{bmatrix} + \begin{bmatrix} B & C \\ C & 0 \end{bmatrix},$$

which corresponds to the vector $v = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$. One eigenvalue of this pencil is $\lambda \approx 4.105\cdot 10^4$. If we first scale Q using the left and right eigenvectors associated with λ and then linearize, this eigenvalue has the condition number $1.2\cdot 10^9$ for the linearization. If we first linearize the QEP and then scale the pencil $L(\lambda)$ using the left and right eigenvectors of λ for the linearization, this eigenvalue has the condition number 5.2. The ratio between the condition numbers is in magnitude what we would expect from applying Theorem 4.2.

However, Theorem 4.2 can be a large overestimate. Assume that $P(\lambda)$ is already almost optimally scaled in the sense of Theorem 3.3, that is, $|y| = |x| = e$ for the left and right eigenvectors y and x associated with the simple finite eigenvalue λ of P. Let $L(\lambda) = \lambda X + Y$ be a linearization of P and let D_1 and D_2 be scaling matrices for L such that $|D_1^{-*}\tilde{y}| = |D_2^{-1}\tilde{x}| = e$ for the left and right eigenvectors $\tilde{y}$ and $\tilde{x}$ of L associated with the eigenvalue λ. The ratio of the condition numbers of the eigenvalue λ for the two pencils L and $D_1 L D_2$ is given by

$$\frac{\kappa_L(\lambda)}{\kappa_{D_1 L D_2}(\lambda)} = \frac{\|\tilde{x}\|_2\|\tilde{y}\|_2}{\|D_1^{-*}\tilde{y}\|_2\|D_2^{-1}\tilde{x}\|_2} \frac{|\lambda|\|X\|_2 + \|Y\|_2}{|\lambda|\|D_1 X D_2\|_2 + \|D_1 Y D_2\|_2}. \tag{4.6}$$

If $L(\lambda) \in \mathbb{DL}(P)$ (4.6) simplifies to

$$\frac{\kappa_L(\lambda)}{\kappa_{D_1 L D_2}(\lambda)} = \frac{1}{\ell}\left(\frac{|\lambda|^{2\ell}-1}{|\lambda|^2-1}\right) \frac{|\lambda|\|X\|_2 + \|Y\|_2}{|\lambda|\|D_1 X D_2\|_2 + \|D_1 Y D_2\|_2}$$

since $|\tilde{x}| = |\tilde{y}| = |\Lambda \otimes e|$. This shows that for $|\lambda| > 1$ the upper bound in Theorem 4.2 can be attained only if

$$\frac{|\lambda|\|X\|_2 + \|Y\|_2}{|\lambda|\|D_1 X D_2\|_2 + \|D_1 Y D_2\|_2} =: \tau(\lambda) \tag{4.7}$$

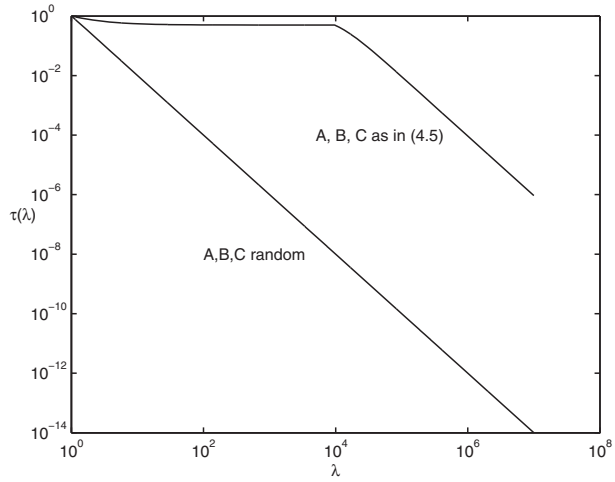


FIG. 4.1. *The function $\tau(\lambda)$ for a large range of values in the case of a random 2×2 QEP and the QEP from* (4.5).

is approximately constant in the range of the eigenvalues in which we are interested. For $L(\lambda) \in \mathbb{DL}(P)$ the matrices D_1 and D_2 are given as

$$D_1 = D_2 = \begin{bmatrix} |\lambda|^{\ell-1} I & & \\ & \ddots & \\ & & I \end{bmatrix} = \operatorname{diag}(|\Lambda|) \otimes I.$$

It follows that for $|\lambda|$ large enough

$$\tau(\lambda) \sim \gamma |\lambda|^{2-2\ell} \tag{4.8}$$

for some constant $\gamma > 0$ and therefore

$$\frac{\kappa_L(\lambda)}{\kappa_{D_1 L D_2}(\lambda)} \sim \frac{\gamma}{\ell}$$

in that case.

In particular, if the upper left $n \times n$ block of X is in norm comparable or larger than the other $n \times n$ subblocks of X, we expect a good agreement of the asymptotic in (4.8) for all $|\lambda| > 1$, where γ is not much larger than 1. Only if the $n \times n$ subblocks of X and Y are of widely varying norm it is possible that $\tau(\lambda)$ is approximately constant for a large range of values of λ, leading to the worst case bound in (4.2) being attained.

The situation is demonstrated in Figure 4.1. For a random 2×2 QEP $\tau(\lambda)$ decays like $\gamma |\lambda|^{-2}$, where $\gamma \approx 1$. For the QEP from (4.5) the function $\tau(\lambda)$ is almost constant for a long time, leading to the worst case bound of Theorem 4.2 being attained in these range of values. Then at about 10^4 it starts decaying like $\gamma |\lambda|^{-2}$, where this time γ is in the order of 10^8.

One of the most frequently used linearizations for unstructured problems is the companion form (4.1). Unfortunately, we cannot immediately apply the previous results to it since the companion form is not in $\mathbb{DL}(P)$ but only in $\mathbb{L}_1(P)$. However, we can still compare the ratio in (4.6). Consider the QEP $Q(\lambda) = \lambda^2 A + \lambda B + C$.

The companion linearization takes the form

$$C_1(\lambda) = \lambda \begin{bmatrix} A & \\ & I \end{bmatrix} + \begin{bmatrix} B & C \\ -I & 0 \end{bmatrix}.$$

We assume that for the left and right eigenvectors y and x associated with the eigenvalue λ of Q we have $|y| = |x| = e$. Furthermore, let D_1 and D_2 again be chosen such that $|D_1^{-*}\tilde{y}| = |D_2^{-1}\tilde{x}| = e$, where $\tilde{y}$ and $\tilde{x}$ are the corresponding left and right eigenvectors for the eigenvalue λ of the companion linearization $C_1(\lambda) = \lambda X + Y$. The relationship between the eigenvectors of C_1 and the eigenvectors of P associated with a finite nonzero eigenvalue λ is given by

$$\tilde{x} = \Lambda \otimes x, \quad \tilde{y} = \begin{bmatrix} y \\ -\frac{1}{\lambda} C^* y \end{bmatrix}.$$

The formula for the left eigenvector is a consequence of [6, Theorem 3.2]. It follows that

$$\frac{\kappa_{C_1}(\lambda)}{\kappa_{\tilde{D}_1 C_1 \tilde{D}_2}(\lambda)} = \frac{1}{2n^{1/2}} \left(\frac{|\lambda|^4 - 1}{|\lambda|^2 - 1} \right)^{1/2} \left(n + \frac{1}{|\lambda|^2} \|C^* y\|_2^2 \right)^{1/2} \tau(\lambda).$$

If $|\lambda| \gg 1$ this simplifies to

$$\frac{\kappa_{C_1}(\lambda)}{\kappa_{\tilde{D}_1 C_1 \tilde{D}_2}(\lambda)} \approx \frac{1}{2} |\lambda| \tau(\lambda),$$

which differs by a factor of $|\lambda|$ from the corresponding case using a $\mathbb{DL}(P)$ linearization. Asymptotically, we have

$$\tau(\lambda) \sim \gamma |\lambda|^{-1}, \quad |\lambda| \gg 1$$

for some factor γ and therefore $\frac{\kappa_{C_1}(\lambda)}{\kappa_{\tilde{D}_1 C_1 \tilde{D}_2}(\lambda)} \sim \frac{\gamma}{2}$, where again we expect this asymptotic to hold approximately for all $|\lambda| > 1$ with a value of γ that is not much larger than 1 if the $n \times n$ subblocks of X and Y do not differ too widely in norm.

5. A heuristic scaling strategy. For standard eigenvalue problems the motivation of scaling algorithms is based on the observation that in floating point arithmetic computed eigenvalues of a matrix A can be expected to be at least perturbed by an amount of the order of $\epsilon_{mach}\|A\|$. Hence, by reducing $\|A\|$ one hopes to reduce the inaccuracies in the computed eigenvalues.

One way of minimizing $\|A\|$ is to find a nonsingular diagonal matrix D such that the rows and columns of A are balanced in the sense that

$$\|D^{-1} A D e_i\| = \|e_i^* D^{-1} A D\|, \quad i = 1, \ldots, n. \tag{5.1}$$

Osborne [14] shows that if A is irreducible and $\|\cdot\|$ is the 2-norm in (5.1), then for this D it holds that

$$\left\| D^{-1} A D \right\|_F = \inf_{\hat{D} \in \mathcal{D}_n} \left\| \hat{D}^{-1} A \hat{D} \right\|_F.$$

A routine that attempts to find a matrix D that balances the row and column norms of A is built into LAPACK under the name `xGEBAL`. It uses the 1-norm in the balancing condition (5.1). A description of the underlying algorithm is contained in [15].

For generalized eigenvalue problems $Ax = \lambda Bx$, Ward [18] proposes to find nonsingular diagonal scaling matrices D_1 and D_2 such that the elements of the scaled matrices D_1AD_2 and D_1BD_2 would have absolute values close to unity. Then the relative perturbations in the matrix elements caused by computational errors would be of similar magnitude. To achieve this Ward proposes to minimize the function

$$\sum_{i,j=1}^{n} (r_i + c_j + \log|A_{ij}|)^2 + (r_i + c_j + \log|B_{ij}|)^2,$$

where the r_i and c_j are the logarithms of the absolute values of the diagonal entries of D_1 and D_2. The scaling by Ward can fail if the matrices A and B contain tiny entries that are not due to bad scaling [10, Example 2.16].

A different strategy for generalized eigenvalue problems is proposed by Lemonnier and Van Dooren [11]. By introducing the notion of generalized normal pencils they motivate a scaling strategy that aims to find nonsingular diagonal matrices D_1 and D_2 such that

$$\|D_1AD_2e_j\|_2^2 + \|D_1BD_2e_j\|_2^2 = \|e_i^*D_1AD_2\|_2^2 + \|e_i^*D_1BD_2\|_2^2 = 1, \quad i,j = 1,\dots,n. \tag{5.2}$$

The scaling condition (5.2) can be generalized in a straightforward way to matrix polynomials of higher degree by

$$\sum_{k=0}^{\ell} \omega^{2k}\|D_1A_kD_2e_i\|_2^2 = 1, \quad \sum_{k=0}^{\ell} \omega^{2k}\|e_j^*D_1A_kD_2\|_2^2 = 1, \quad i,j = 1,\dots,n \tag{5.3}$$

for some $\omega > 0$ that is chosen in magnitude close to the wanted eigenvalues. The intuitive idea behind (5.3) is to balance rows and columns of the coefficient matrices A_k while taking into account the weighting of the coefficient matrices induced by the eigenvalue parameter; that is, for very large eigenvalues the rows and columns of A_ℓ dominate and for very small eigenvalues the rows and columns of A_0 dominate. This also reflects the result of Theorem 3.3 that the optimal scaling matrices are dependent on the wanted eigenvalue. In section 7 we show that including the estimate ω can greatly improve the results of scaling.

In [11] Lemonnier and Van Dooren introduced a linearly convergent iteration to obtain matrices D_1 and D_2 consisting of powers of 2 that approximately satisfy (5.2). The idea in their code is to alternatively update D_1 and D_2 by first normalizing all rows of $\begin{bmatrix} A & B \end{bmatrix}$ and then all columns of $\begin{bmatrix} A \\ B \end{bmatrix}$. The algorithm repeats this operation until (5.2) is approximately satisfied. This iteration can easily be extended to weighted scaling of matrix polynomials. This is done in Algorithm 1. The main difference to the MATLAB code in [11] is the definition of the variable M in line 6 that now accommodates matrix polynomials and the weighting parameter ω.

If we do not have any estimate for the magnitude of the wanted eigenvalues, a possible choice is to set $\omega = 1$ in (5.3). In that case all coefficient matrices have the same weight in that condition.

6. Transformations of the eigenvalue parameter. In the previous sections we investigated how diagonal scaling of $P(\lambda)$ by multiplication of $P(\lambda)$ with left and right scaling matrices $D_1, D_2 \in \mathcal{D}_n$ can improve the condition number of the eigenvalues. In this section we consider scaling a PEP by transforming the eigenvalue parameter λ. This was proposed by Fan, Lin, and Van Dooren for quadratics in

ALGORITHM 1 Diagonal scaling of $P(\lambda) = \lambda^\ell A_\ell + \cdots + \lambda A_1 + A_0$.

Require: $A_0, \ldots, A_\ell \in \mathbb{C}^{n\times n}$, $\omega > 0$.
1: $M \leftarrow \sum_{k=0}^{\ell} |\lambda|^{2k} |A_k|.^2$, $D_1 \leftarrow I$, $D_2 \leftarrow I$ ($|A_k|.^2$ is entry-wise square)
2: $maxiter \leftarrow 5$
3: **for** $iter = 1$ to $maxiter$ **do**
4: $emax \leftarrow 0$, $emin \leftarrow 0$
5: **for** $i = 1$ to n **do**
6: $d \leftarrow \sum_{j=0}^{n} M(i,j)$, $e \leftarrow -\text{round}(\frac{1}{2}\log_2 d)$
7: $M(i,:) \leftarrow 2^{2e} \cdot M(i,:)$, $D_1(i,i) \leftarrow 2^e \cdot D_1(i,i)$
8: $emax \leftarrow \max(emax, e)$, $emin \leftarrow \min(emin, e)$
9: **end for**
10: **for** $i = 1$ to n **do**
11: $d \leftarrow \sum_{j=0}^{n} M(j,i)$, $e \leftarrow -\text{round}(\frac{1}{2}\log_2 d)$
12: $M(:,i) \leftarrow 2^{2e} \cdot M(:,i)$, $D_2(i,i) \leftarrow 2^e \cdot D_2(i,i)$
13: $emax \leftarrow \max(emax, e)$, $emin \leftarrow \min(emin, e)$
14: **end for**
15: **if** $emax \leq emin + 2$ **then**
16: **BREAK**
17: **end if**
18: **end for**
19: **return** D_1, D_2

[3] (see also [6]). Let $Q(\lambda) := \lambda^2 A_2 + \lambda A_1 + A_0$. Define the quadratic polynomial $\widetilde{Q}(\mu) = \mu^2 \widetilde{A}_2 + \mu \widetilde{A}_1 + \widetilde{A}_0$ as

$$\widetilde{Q}(\mu) := \beta Q(\alpha\mu) = \beta\mu^2\alpha^2 A_2 + \beta\mu\alpha A_1 + \beta A_0.$$

The parameters $\beta > 0$ and $\alpha > 0$ are chosen such that the 2-norms of the new coefficient matrices $\widetilde{A}_2 := \beta\alpha^2 A_2$, $\widetilde{A}_1 := \beta\alpha A_1$, and $\widetilde{A}_0 := \beta A_0$ are as close to 1 as possible; that is, we need to solve

$$\min_{\alpha>0,\beta>0} \max\left\{|\beta\alpha^2\|A_2\|_2 - 1|, |\beta\alpha\|A_1\|_2 - 1|, |\beta\|A_0\|_2 - 1|\right\}. \tag{6.1}$$

It is shown in [3] that the unique solution of (6.1) is given by

$$\alpha = \left(\frac{\|A_0\|_2}{\|A_2\|_2}\right)^{\frac{1}{2}}, \quad \beta = \frac{2}{\|A_0\|_2 + \|A_1\|_2\alpha}.$$

Hence, after scaling we have $\|\widetilde{A}_0\|_2 = \|\widetilde{A}_2\|_2$. The motivation behind this scaling is that solving a QEP by applying a backward stable algorithm to solve (4.1) is backward stable if $\|A_0\|_2 = \|A_1\|_2 = \|A_2\|_2$ [17, Theorem 7]. For matrix polynomials of arbitrary degree ℓ it is shown in [7] that with

$$\rho := \frac{\max_i \|A_i\|_2}{\min(\|A_0\|_2, \|A_\ell\|_2)} \geq 1$$

one has

$$\frac{2\sqrt{\ell}}{\ell+1}\frac{1}{\rho} \leq \frac{\inf_v \kappa_L(\lambda; v; P)}{\kappa_P(\lambda)} \leq \ell^2\rho,$$

where $\kappa_L(\lambda; v; P)$ is the condition number of the eigenvalue λ for the linearization $L(\lambda) \in \mathbb{DL}(P)$ with vector v. Hence, if $\rho \approx 1$ then there is $L(\lambda) \in \mathbb{DL}(P)$ such that $\kappa_L(\lambda; v; P) \approx \kappa_P(\lambda)$. For backward errors analogous results were shown in [6]. The aim is therefore to find a transformation of λ such that ρ is minimized. For the transformation $\lambda = \alpha\mu$ the solution is given in the following theorem.

THEOREM 6.1. [illegible] $P(\lambda)$ [illegible] ℓ [illegible]

$$\rho(\alpha) := \frac{\max_{0\le i\le \ell} \alpha^i \|A_i\|_2}{\min(\|A_0\|_2, \alpha^\ell \|A_\ell\|_2)}$$

[illegible] $\alpha > 0$ [illegible] $\rho(\alpha)$ [illegible] $\alpha_{opt} = (\|A_0\|_2/\|A_\ell\|_2)^{\frac{1}{\ell}}$ [illegible]. The function $\rho(\alpha)$ is continuous. Furthermore, for $\alpha \to 0$ and $\alpha \to \infty$ we have $\rho(\alpha) \to \infty$. Hence, there must be at least one minimium in $(0, \infty)$. Let $\tilde{\alpha}$ be a local minimizer. Now assume that $\|A_0\|_2 < \tilde{\alpha}^\ell \|A_\ell\|_2$. Then

$$\rho(\alpha) = \frac{1}{\|A_0\|_2} \max(\alpha\|A_1\|_2, \ldots, \alpha^\ell \|A_\ell\|_2)$$

in a neighborhood of $\tilde{\alpha}$. But this function is strictly increasing in this neighborhood. Hence, $\tilde{\alpha}$ cannot be a minimizer. Similarly, the assumption $\|A_0\|_2 > \tilde{\alpha}^\ell \|A_\ell\|_2$ at the minimum leads to

$$\rho(\alpha) = \frac{1}{\alpha^\ell \|A_\ell\|_2} \max\left(\|A_0\|_2, \ldots, \alpha^{\ell-1}\|A_{\ell-1}\|_2\right),$$

in a neighborhood of this minimum, which is strictly decreasing. A necessary condition for a minimizer is therefore given as $\|A_0\|_2 = \alpha^\ell \|A_\ell\|_2$, which has the unique solution $\alpha_{opt} = (\|A_0\|_2/\|A_\ell\|_2)^{\frac{1}{\ell}}$ in $(0, \infty)$. Since there must be at least one minimum of $\rho(\alpha)$ in $(0, \infty)$, it follows that α_{opt} is the unique minimizer there. ∎

We emphasize that the variable transformation $\lambda = \alpha\mu$ does not change condition numbers or backward errors of the original polynomial problem. It affects only these quantities for the linearization $L(\lambda)$.

For the special case $\ell = 2$, this leads to the same scaling as proposed by Fan, Lin, and Van Dooren [3]. If $\|A_0\|_2 = \|A_\ell\|_2$, then $\alpha_{opt} = 1$ and we cannot improve ρ with the transformation $\lambda = \alpha\mu$. In that case one might consider more general Möbius transformations of the type

$$\widetilde{P}(\mu) := (c\mu + d)^\ell P\left(\frac{a\mu + b}{c\mu + d}\right), \quad a, b, c, d \in \mathbb{C}.$$

However, it is still unclear how to choose the parameters a, b, c, d in order to improve ρ for a specific matrix polynomial.

7. Numerical examples. We first present numerical experiments on sets of randomly generated PEPs. The test problems are created by defining $A_k := F_1^{(k)} \tilde{A}_k F_2^{(k)}$, where the entries of $\tilde{A}_k$ are $N(0, 1)$ distributed random numbers and the entries of $F_1^{(k)}$ and $F_2^{(k)}$ are jth powers of $N(0, 1)$ distributed random numbers obtained from the `randn` function in MATLAB. As j increases these matrices become more badly scaled and ill-conditioned. This is a similar strategy to create test matrices as was used in [11]. In our experiments we choose the parameter $j = 6$.

In Figure 7.1(a) we show the ratio of the normwise and componentwise eigenvalue condition numbers of the eigenvalues for 100 quadratic eigenvalue problems of

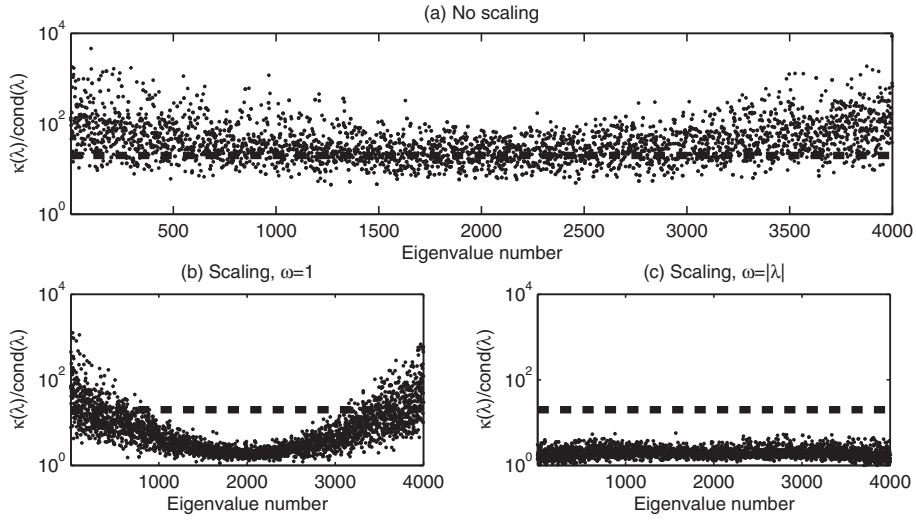


FIG. 7.1. (a) *The ratio of the normwise and componentwise condition numbers for the eigenvalues of* 100 *randomly created quadratic test problems of dimension* $n = 20$ *before scaling.* (b) *The same test set but now after scaling with* $\omega = 1$. (c) *Eigenvalue dependent scaling with* $\omega = |\lambda|$.

dimension $n = 20$. The eigenvalues range in magnitude from 10^{-8} to 10^{8} and are sorted in ascending magnitude. According to Theorem 3.3 the ratio of normwise and componentwise condition number is smaller than n (shown by the dotted line) if the problem is almost optimally scaled for the corresponding eigenvalue. But only a few eigenvalues satisfy this condition. Hence, we expect that scaling will improve the normwise condition numbers of the eigenvalues in these test problems. In Figure 7.1(b) the test problems are scaled using Alg. 1 with the fixed parameter $\omega = 1$. Apart from the extreme ones, all eigenvalues are now almost optimally scaled. In Figure 7.1(c) an eigenvalue dependent scaling is used; that is, $\omega = |\lambda|$ for each eigenvalue λ. Now all eigenvalues are almost optimally scaled. This demonstrates that having some information about the magnitude of the wanted eigenvalues can greatly improve the results of scaling.

The source of badly scaled eigenvalue problems often lies in a nonoptimal choice of units in the modelling process, which can lead to all coefficient matrices A_k being badly scaled in a similar way. In that case it is not necessary to provide any kind of weighting. This is demonstrated by the example in Figure 7.2. The left plot in that figure shows the ratio of the normwise and componentwise condition numbers of the eigenvalues of another set of eigenvalue problems. Again, we choose $n = 20$ and $\ell = 2$. However, this time the matrices $F_1^{(k)}$ and $F_2^{(k)}$ in the definition $A_k := F_1^{(k)} \tilde{A}_k F_2^{(k)}$ are kept constant for all $k = 0, \ldots, \ell$. They vary only between different eigenvalue test problems. The right plot in Figure 7.2 shows the ratio of normwise and componentwise condition number after scaling using $\omega = 1$. Now all eigenvalue condition numbers are almost optimal.

Let us now consider the example of a 4th order PEP $(\lambda^4 A_4 + \lambda^3 A_3 + \lambda^2 A_2 + \lambda A_1 + A_0)x = 0$ derived from the Orr–Sommerfeld equation [16]. The matrices are created with the NLEVP benchmark collection [1]. To improve the scaling factor ρ, we substitute $\lambda = \mu\alpha_{opt}$, where $\alpha_{opt} \approx 8.42 \cdot 10^{-4}$. This reduces ρ from $1.99 \cdot 10^{12}$ to 4.86. The ratio $\kappa_P(\mu)/\mathrm{cond}_P(\mu)$ for the unscaled problem is shown in Figure 7.3(a).

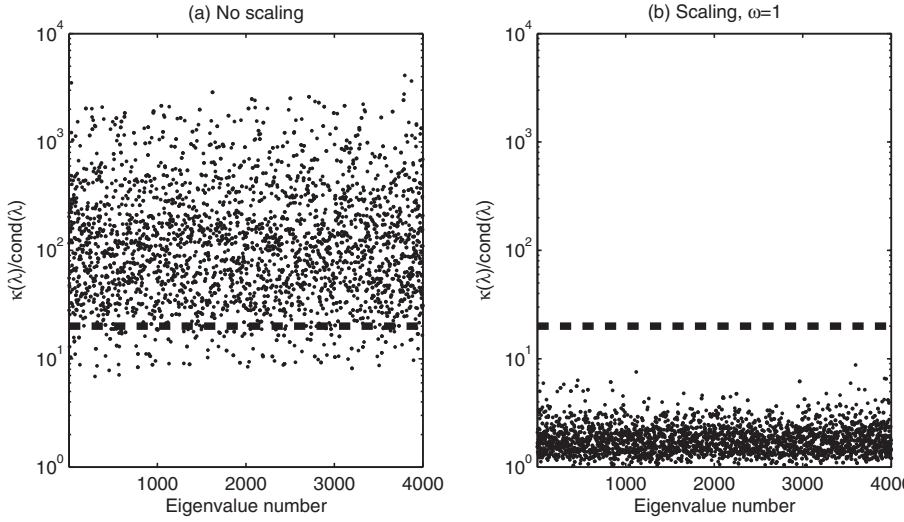


FIG. 7.2. *In this test all coefficient matrices of an eigenvalue problem are badly scaled in a similar way.* (a) *Ratio of normwise and componentwise condition condition numbers before scaling.* (b) *The same ratio after scaling with* $\omega = 1$.

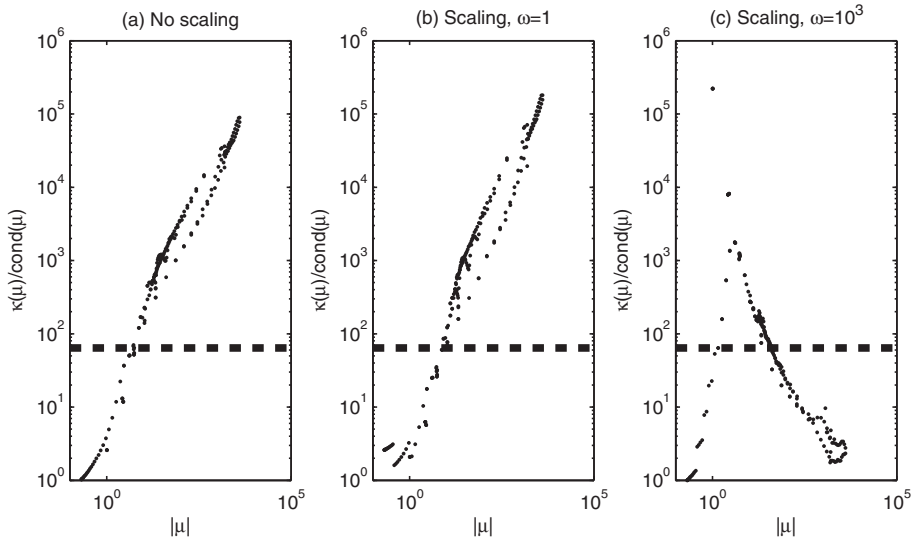


FIG. 7.3. *Scaling of a 4th order PEP.* (a) $\kappa(\mu)/\mathrm{cond}(\mu)$ *for the unscaled PEP.* (b) *The same ratio after scaling with* $\omega = 1$. (c) *Scaling with* $\omega = 10^3$. *The horizontal lines denote the dimension* $n = 64$ *of the PEP.*

The x-axis denotes the absolute value $|\mu|$ of an eigenvalue μ. The horizontal line shows the dimension $n = 64$ of the problem. The large eigenvalues in this problem are far away from being optimally scaled. In Figure 7.3(b) we use Alg. 1 with the weighting parameter $\omega = 1$. This has almost no effect on the normwise condition numbers of the eigenvalues. In Figure 7.3(c) we use $\omega = 10^3$. Now the larger eigenvalues are almost optimally scaled while the normwise condition numbers of some of the smaller eigenvalues have become worse. Hence, in this example the right choice of

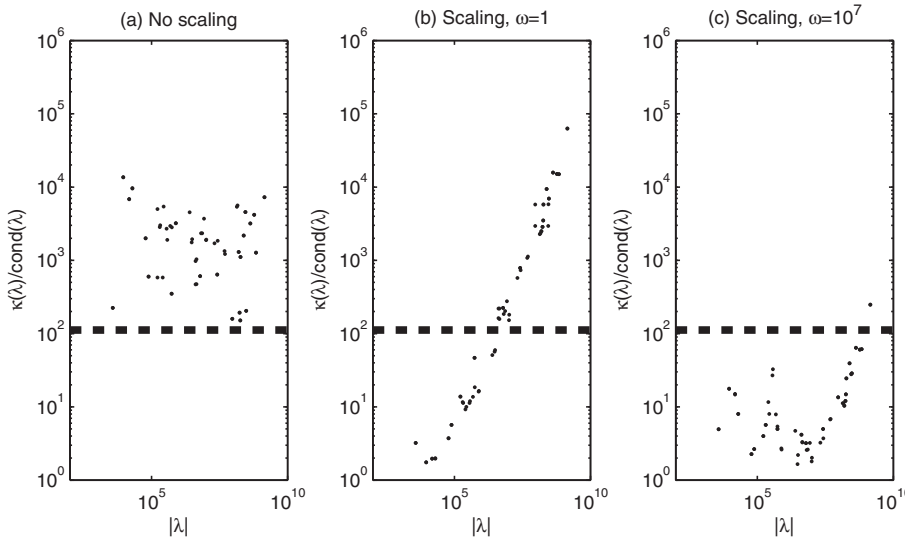


FIG. 7.4. *Scaling of the GEP $Kx = \lambda Mx$, where K and M are the matrices BCSSTK*03 *and BCSSTM*03 *from Matrix Market* [13]. *The best overall results are obtained with $\omega = 10^7$ (see right graph). The horizontal lines denote the dimension $n = 112$ of the GEP.*

the weighting parameter ω is crucial. If we want to improve the scaling of the large eigenvalue, we need to choose ω as approximately the magnitude of these values to obtain good results. By diagonal scaling with D_1 and D_2 the scaling factor ρ might increase again. In this example, after diagonal scaling using the weight $\omega = 10^3$, ρ increases to $1.8 \cdot 10^5$. However, we can reduce this again by another variable transformation of the form $\mu = \tilde{\alpha}_{opt}\tilde{\mu}$. From Theorem 6.1 it follows that $\tilde{\alpha}_{opt} \approx 13.9$, and after this variable transformation ρ reduces to 67.6. Hence, at the end the condition numbers of the largest eigenvalues have decreased by a factor of about 10^5, while the scaling factor ρ has increased only by a factor of about 10.

Not only for polynomial problems can a weighted scaling significantly improve the condition numbers compared to unweighted scaling. In Figure 7.4 we show the results of scaling for the GEP $Kx = \lambda Mx$, where K and M are the matrices BCSSTK03 and BCSSTM03 from Matrix Market [13]. The dimension of the GEP is 112. While unweighted scaling improves the condition number of the smaller eigenvalues, the best result is obtained by using the weighting parameter $\omega = 10^7$. Then the condition number of all eigenvalues is improved considerably.

8. Some remarks about scaling in practice. In this concluding section we want to give some suggestions for practical scaling algorithms based on the results of this paper.

1. [illegible] $\kappa(\lambda)$ [illegible] $\mathrm{cond}(\lambda)$ [illegible] At the moment eigensolvers often return a normwise condition number if desired by the user. It is only a little more effort to additionally compute the ratio $\kappa(\lambda)/\mathrm{cond}(\lambda)$. From Theorem 3.3 it follows that a polynomial is almost optimally scaled for a certain eigenvalue if $\kappa(\lambda)/\mathrm{cond}(\lambda) \leq n$. If this condition is violated, the user may decide to rescale the eigenvalue problem and then to recompute the eigenvalues in order to improve their accuracy.

2. [illegible] The numerical examples in section 7 show that the results of scaling can be greatly improved if ω is chosen to be of the magnitude of

the wanted eigenvalues. In many applications this information is available from other considerations. If no information about the eigenvalues is available, a reasonable choice is to set $\omega = 1$.

3. [illegible] The results in section 4 show that one can obtain a smaller condition number if one scales after linearizing the polynomial $P(\lambda)$. If the eigenvalues of the linearization $L(\lambda)$ are computed without taking any special structure of $L(\lambda)$ into account, this is therefore the preferable way. However, if the eigensolver uses the special structure of the linearization $L(\lambda)$, then one should scale the original polynomial $P(\lambda)$ and then linearize in order not to destroy this structure.

4. [illegible] $\lambda = \alpha\mu$ [illegible] ρ This technique, which was introduced by Fan, Lin, and Van Dooren [3] for quadratics and generalized in Theorem 6.1, often reduces the ratio of the condition number of an eigenvalue λ between the linearization and the original polynomial. In practice we would compute α using the Frobenius or another cheaply computable norm.

The first two suggestions also apply to generalized linear eigenvalue problems and can be easily implemented to current standard solvers for them. Further research is needed for the effect of scaling on the backward error. Bounds on the backward error after scaling are difficult to obtain since the computed eigenvalues change after scaling and this change depends on the eigensolver.

Acknowledgments. The author would like to thank Nick Higham and Françoise Tisseur for their support and advice on this work. Furthermore, the author would like to thank the anonymous referees whose comments improved the paper.

REFERENCES

[1] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur, *NLEVP: A collection of nonlinear eigenvalue problems*, MIMS EPrint 2008.40, Manchester Institute of Mathematical Sciences, University of Manchester, 2008.

[2] T.-Y. Chen and J. W. Demmel, *Balancing sparse matrices for computing eigenvalues*, Linear Algebra Appl., 309 (2000), pp. 261–287.

[3] H.-Y. Fan, W.-W. Lin, and P. Van Dooren, *Normwise scaling of second order polynomial matrices*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 252–256.

[4] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Academic Press, New York, 1982.

[5] D. J. Higham and N. J. Higham, *Structured backward error and condition of generalized eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 493–512.

[6] N. J. Higham, R.-C. Li, and F. Tisseur, *Backward error of polynomial eigenproblems solved by linearization*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1218–1241.

[7] N. J. Higham, D. S. Mackey, and F. Tisseur, *The conditioning of linearizations of matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1005–1028.

[8] N. J. Higham, *A survey of componentwise perturbation theory in numerical linear algebra*, in Mathematics of Computation 1943–1993: A Half Century of Computational Mathematics, W. Gautschi, ed., Proc. Sympos. Appl. Math. 48, AMS, Providence, RI, 1994, pp. 49–77.

[9] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[10] D. Kressner, *Numerical methods for general and structured eigenvalue problems*, Lecture Notes in Comput. Sci. Engrg. 46, Springer-Verlag, Berlin, 2005.

[11] D. Lemonnier and P. Van Dooren, *Balancing regular matrix pencils*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 253–263.

[12] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 971–1004.

[13] *Matrix Market*. http://math.nist.gov/MatrixMarket/.

[14] E. E. Osborne, *On pre-conditioning of matrices*, J. ACM, 7 (60), pp. 338–345.

[15] B. N. Parlett and C. Reinsch, *Balancing a matrix for calculation of eigenvalues and eigenvectors*, Numer. Math., 13 (1969), pp. 293–304.
[16] F. Tisseur and N. J. Higham, *Structured pseudospectra for polynomial eigenvalue problems, with applications*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 187–208.
[17] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361.
[18] R. C. Ward, *Balancing the generalized eigenvalue problem*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 141–152.
[19] D. S. Watkins, *A case where balancing is harmful*, Electron. Trans. Numer. Anal., 23 (2006), pp. 1–4.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1339–1357

EXPLICIT SOLUTIONS FOR A RICCATI EQUATION FROM TRANSPORT THEORY*

VOLKER MEHRMANN† AND HONGGUO XU‡

In memoriam of Gene H. Golub

Abstract. We derive formulas for the minimal positive solution of a particular nonsymmetric Riccati equation arising in transport theory. The formulas are based on the eigenvalues of an associated matrix. We use the formulas to explore some new properties of the minimal positive solution and to derive fast and highly accurate numerical methods. Some numerical tests demonstrate the properties of the new methods.

Key words. nonsymmetric Riccati equation, secular equation, eigenvalues, minimal positive solution, Cauchy matrix, transport theory, quadrature formula

AMS subject classifications. 15A24, 65F15, 82C70, 65H05

DOI. 10.1137/070708743

1. Introduction.

We consider nonsymmetric matrix Riccati equations of the special form

$$XA + DX - XBX - C = 0, \tag{1.1}$$

with

$$A = \Gamma - pe^T, \quad D = \Delta - ep^T, \quad B = pp^T, \quad C = ee^T,$$

where

$$\Gamma := \operatorname{diag}(\gamma_1, \ldots, \gamma_n), \quad \Delta := \operatorname{diag}(\delta_1, \ldots, \delta_n),$$
$$p = [p_1, \ldots, p_n]^T, \quad e = [1, \ldots, 1]^T,$$

and $\gamma_n > \cdots > \gamma_1 > 0$, $\delta_n > \cdots, \delta_1 > 0$, and $p_1, \ldots, p_n > 0$.

Such Riccati equations arise in Markov models [28] and in nuclear physics [8, 18, 22]. In the latter application, to study the transport of particles, one introduces integral equations of the form

$$\left[\frac{1}{x+\alpha} + \frac{1}{y-\alpha}\right] T(x,y) = \beta \left[1 + \frac{1}{2}\int_{-\alpha}^{1} \frac{T(t,y)}{t+\alpha} dt\right] \left[1 + \frac{1}{2}\int_{\alpha}^{1} \frac{T(x,t)}{t-\alpha} dt\right], \tag{1.2}$$

where the unknown function $T(x,y) : [-\alpha, 1] \times [\alpha, 1] \mapsto \mathbb{R}^+$ is called the [illegible], $\alpha \in [0, 1)$ is an angular shift, and $\beta \in [0, 1]$ is the average of the total

*Received by the editors November 20, 2007; accepted for publication (in revised form) by J. H. Brandts June 4, 2008; published electronically October 16, 2008.
http://www.siam.org/journals/simax/30-4/70874.html

†Institut für Mathematik, TU Berlin, Str. des 17. Juni 136, D-10623 Berlin, Germany (mehrmann@math.tu-berlin.de). This author's research was partially supported by Deutsche Forschungsgemeinschaft, through the DFG Research Center MATHEON Mathematics for Key Technologies in Berlin.

‡Department of Mathematics, University of Kansas, Lawrence, KS 44045 (xu@math.ku.edu). This author's research was partially supported by the University of Kansas General Research Fund allocation # 2301717 and by Deutsche Forschungsgemeinschaft through the DFG Research Center MATHEON Mathematics for Key Technologies in Berlin.

number of particles emerging from a collision. (Here $\mathbb{R}^+$ denotes the set of positive real numbers.)

To solve this integral equation numerically, one approximates the integrals via classical quadrature formulas [29]. For this the function $T(x, y)$ is approximated via a matrix $X = [x_{ij}]$, where x_{ij} is an approximation of $T(\mu_i, \nu_j)$ with μ_i, ν_j being the ith and jth nodes of the quadrature formula on $[-\alpha, 1]$ and $[\alpha, 1]$, respectively; see, e.g., [18].

In this discretization the matrix X has to satisfy the matrix Riccati equation (1.1) with coefficient matrices

$$\gamma_j = \frac{1}{\beta(1-\alpha)\omega_j}, \quad \delta_j = \frac{1}{\beta(1+\alpha)\omega_j}, \quad p_j = \frac{c_j}{2\omega_j}, \tag{1.3}$$

for $j = 1, 2, \ldots, n$, where $\{c_j\}_{j=1}^n$, $\{w_j\}_{j=1}^n$ are the sets of weights and nodes of the specific quadrature rule that is used on the interval [0, 1]. These typically satisfy

$$c_1, \ldots, c_n > 0, \quad \sum_{j=1}^{n} c_j = 1; \qquad 1 > \omega_1 > \cdots > \omega_n > 0. \tag{1.4}$$

In [20] it is shown that the Riccati equation (1.1) has two entrywise positive solutions $X = [x_{ij}], Y = [y_{ij}] \in \mathbb{R}^{n,n}$, which satisfy $X \leq Y$, where we use the notation that $X \leq Y$ if $x_{ij} \leq y_{ij}$ for all $i, j = 1, \ldots, n$.

In the applications from transport theory, only X, the smaller of the two positive solutions, is of interest. Therefore, in this paper we consider only the computation of the minimal positive solution X. The computation of this minimal solution has been investigated in several publications. Various direct and iterative methods [1, 2, 12, 13, 14, 15, 16, 17, 18, 19, 25] have been proposed by either directly solving the Riccati equation or by computing specific invariant subspaces of the $2n \times 2n$ matrix

$$H = \begin{bmatrix} A & -B \\ C & -D \end{bmatrix} \tag{1.5}$$

that is formed from the coefficient matrices.

In [20] even an explicit solution formula has been derived that is based on the eigenvalues H. Motivated by this result, we derive different explicit formulas, one of which is mathematically equivalent to the one in [20], but of a much simpler form. We will use these formulas to derive both entrywise and normwise bounds for the solution matrix and show that the entries of the solution have a graded entry property. We will also use the formulas to develop fast and highly accurate numerical algorithms for the minimal positive solution of (1.1).

This paper is organized as follows. In section 2, we will reformulate the associated eigenvalue problem via an appropriate balancing strategy. We use the associated secular function to derive some properties of the eigenvalues of H. In section 3, we then derive four formulas for the minimal positive solution based on the eigenvalues. Entrywise and normwise bounds for the minimal positive solution are provided in section 4. Numerical algorithms and an error analysis are presented in section 5 and some numerical examples are shown in section 6. A conclusion is given in section 7.

Throughout this paper, $\lambda(A)$ denotes the spectrum of a square matrix A, and I_n (or simply I) is the $n \times n$ identity matrix. The norm used in this paper is the spectral norm.

2. Spectral properties of the matrix H. In this section we analyze the spectral properties of the matrix H in (1.5) defined by the coefficient matrices of (1.1).

In order for all of the eigenvalues of H to be real, we assume that the condition

$$1 - \sum_{j=1}^{n} p_j \left(\frac{1}{\gamma_j} + \frac{1}{\delta_j} \right) \geq 0 \tag{2.1}$$

holds. The transport problem with the coefficients defined in (1.3) and (1.4) is a special case where this assumption is satisfied.

The first step in our analysis is a balancing of the coefficient matrices. Since the entries of the vector p are positive, we may define

$$\Phi := \operatorname{diag}(\sqrt{p_1}, \ldots, \sqrt{p_n}), \qquad \phi := [\sqrt{p_1}, \ldots, \sqrt{p_n}]^T.$$

Using Φ to scale the Riccati equation (1.1) via

$$\begin{aligned}
\tilde{X} &= \Phi X \Phi, \\
\tilde{A} &= \Phi^{-1} A \Phi = \Gamma - \phi\phi^T, \\
\tilde{D} &= \Phi D \Phi^{-1} = \Delta - \phi\phi^T, \\
\tilde{B} &= \Phi^{-1} B \Phi^{-1} = \phi\phi^T, \\
\tilde{C} &= \Phi C \Phi = \phi\phi^T = \tilde{B},
\end{aligned}$$

we obtain the equivalent Riccati equation

$$\tilde{X}\tilde{A} + \tilde{D}\tilde{X} - \tilde{X}\tilde{B}\tilde{X} - \tilde{B} = 0, \tag{2.2}$$

and obviously, X is a solution to (1.1) if and only if $\tilde{X} = \Phi X \Phi$ is a solution to (2.2). For the associated matrix formed from the coefficients we then have

$$\begin{aligned}
\tilde{H} &= \begin{bmatrix} \Phi^{-1} & 0 \\ 0 & \Phi \end{bmatrix} H \begin{bmatrix} \Phi & 0 \\ 0 & \Phi^{-1} \end{bmatrix} \\
&= \begin{bmatrix} \tilde{A} & -\tilde{B} \\ \tilde{B} & -\tilde{D} \end{bmatrix} = \begin{bmatrix} \Gamma & 0 \\ 0 & -\Delta \end{bmatrix} - \begin{bmatrix} \phi \\ -\phi \end{bmatrix} \begin{bmatrix} \phi \\ \phi \end{bmatrix}^T, \qquad (2.3)
\end{aligned}$$

and we see that $\tilde{H}$ is similar to H and is a rank-one modification of a diagonal matrix, which is analogous to the real symmetric rank-one updating problem discussed in [9]. It follows that the eigenvalues of $\tilde{H}$ can be obtained cheaply and accurately via the solution of secular equations by using a method similar to the one discussed in [10, section 8.5].

Furthermore, it is well known (see, e.g., [23]) that $\tilde{X}$ is a solution to (2.2) if and only if $\tilde{X}$ satisfies the invariant subspace equation

$$\tilde{H} \begin{bmatrix} I \\ \tilde{X} \end{bmatrix} = \begin{bmatrix} I \\ \tilde{X} \end{bmatrix} (\tilde{A} - \tilde{B}\tilde{X}).$$

In [20] it was shown (for the original solution X) that $\tilde{X}$ is the minimal positive solution if and only if all of the eigenvalues of $\tilde{A} - \tilde{B}\tilde{X}$ are nonnegative.

In order to analyze the properties of the matrix $\tilde{H}$ and thus also of the similar matrix H, we first derive some properties of the eigenvalues of $\tilde{H}$.

Consider the rational function

$$\chi(\lambda) = 1 + \sum_{j=1}^{n} \frac{p_j}{\lambda - \gamma_j} - \sum_{j=1}^{n} \frac{p_j}{\lambda + \delta_j}. \tag{2.4}$$

Then, since

$$\det(\lambda I - \tilde{H}) = \chi(\lambda) \left(\prod_{j=1}^{n} (\lambda - \gamma_j)(\lambda + \delta_j) \right), \tag{2.5}$$

it follows that the eigenvalues of $\tilde{H}$ are just the roots of the [illegible] $\chi(\lambda) = 0$, and thus the computation of the spectrum of $\tilde{H}$ can be carried out very efficiently by solving the secular equation; see [11, 27]. Furthermore, we have the following interlacing properties.

LEMMA 2.1. [illegible] $\tilde{H}$ [illegible] (2.2) [illegible] (2.1) [illegible] $\tilde{H}$ [illegible] $2n$ [illegible] $-\nu_n < \cdots < -\nu_1 \le 0$, $0 \le \lambda_1 < \cdots < \lambda_n$ [illegible]

$$0 \le \nu_1 < \delta_1 < \nu_2 < \delta_2 < \cdots < \nu_{n-1} < \delta_{n-1} < \nu_n < \delta_n$$

[illegible]

$$0 \le \lambda_1 < \gamma_1 < \lambda_2 < \gamma_2 < \cdots < \lambda_{n-1} < \gamma_{n-1} < \lambda_n < \gamma_n.$$

[illegible]

1. $\nu_1 = 0$ [illegible] $\lambda_1 > 0$, [illegible] $\chi(0) = 0$ [illegible] $\chi'(0) > 0$
2. $\nu_1 > 0$ [illegible] $\lambda_1 = 0$, [illegible] $\chi(0) = 0$ [illegible] $\chi'(0) < 0$
3. $\nu_1 = \lambda_1 = 0$, [illegible] $\chi(0) = \chi'(0) = 0$ [illegible] $\tilde{H}$ [illegible] 2×2 [illegible] 0

[illegible]. The proof is basically given already in [20] based on the properties of the secular function $\chi(\lambda)$. Note that assumption (2.1) implies that $\chi(0) \ge 0$.

The second part of the third case has already been shown in [12] in a more general setting. □

[illegible] 2.2. Suppose the quadrature formula that is used to discretize the integral equation (1.2) is of order greater than or equal to 3; i.e.,

$$\sum_{j=1}^{n} c_j w_j^k = \frac{1}{k+1}, \qquad k = 0, 1, 2, 3.$$

With (1.3) it is easily verified that

$$\chi(0) = 1 - \sum_{j=1}^{n} \left(\frac{p_j}{\gamma_j} + \frac{p_j}{\delta_j} \right) = 1 - \beta \sum_{j=1}^{n} c_j = 1 - \beta,$$

$$\chi'(0) = \sum_{j=1}^{n} \left(-\frac{p_j}{\gamma_j^2} + \frac{p_j}{\delta_j^2} \right) = 2\alpha\beta^2 \sum_{j=1}^{n} c_j w_j = \alpha\beta^2,$$

$$\chi''(0) = -2 \sum_{j=1}^{n} \left(\frac{p_j}{\gamma_j^3} + \frac{p_j}{\delta_j^3} \right) = -2(1+3\alpha^2)\beta^3 \sum_{j=1}^{n} c_j w_j^2 = -\frac{2}{3}(1+3\alpha^2)\beta^3,$$

$$\chi'''(0) = 6 \sum_{j=1}^{n} \left(-\frac{p_j}{\gamma_j^4} + \frac{p_j}{\delta_j^4} \right) = 24\alpha(1+\alpha^2)\beta^4 \sum_{j=1}^{n} c_j w_j^3 = 6\alpha(1+\alpha^2)\beta^4.$$

Since $\chi'(0) \geq 0$, we have that case 1 in Lemma 2.1 happens when $\beta = 1$ and $\alpha > 0$ and case 3 happens when $\beta = 1$ and $\alpha = 0$. Case 2 will never happen.

3. Formulas for the minimal positive solution. In this section we will derive explicit formulas for the minimal positive solution of (1.1) in terms of the eigenvalues $-\nu_1, \ldots, -\nu_n, \lambda_1, \ldots, \lambda_n$ of H (or $\tilde{H}$). For this we need the following lemma.

LEMMA 3.1. [illegible] $\tilde{X} \in \mathbb{R}^{n,n}$ [illegible]

(a) $\tilde{X}$ [illegible] (2.2)

(b) $\tilde{X}$ [illegible]

$$\tilde{H} \begin{bmatrix} I_n \\ \tilde{X} \end{bmatrix} = \begin{bmatrix} I_n \\ \tilde{X} \end{bmatrix} \tilde{R}_1,$$

[illegible] $\tilde{R}_1 = \tilde{A} - \tilde{B}\tilde{X}$ [illegible] $\sigma(\tilde{R}_1) = \{\lambda_1, \ldots, \lambda_n\}$.

(c) $\tilde{X}^T$ [illegible]

$$\tilde{Y}\tilde{D} + \tilde{A}\tilde{Y} - \tilde{Y}\tilde{B}\tilde{Y} - \tilde{B} = 0. \tag{3.1}$$

(d) $\tilde{X}$ [illegible]

$$\tilde{H} \begin{bmatrix} \tilde{X}^T \\ I_n \end{bmatrix} = \begin{bmatrix} \tilde{X}^T \\ I_n \end{bmatrix} \tilde{R}_2, \tag{3.2}$$

[illegible] $\tilde{R}_2 = -(\tilde{D} - \tilde{B}\tilde{X}^T)$ [illegible] $\sigma(\tilde{R}_2) = \{-\nu_1, \ldots, -\nu_n\}$.

[illegible]. The equivalence of (a) and (b) is given in [20]. The equivalence between (a) and (c) is obvious by taking the transpose on both sides of (2.2) or (3.1). The equivalence between (c) and (d) is shown in [12]. □

With formulas for $\tilde{R}_1, \tilde{R}_2$ as in Lemma 3.1 and the formulas for $\tilde{A}$, $\tilde{D}$ and $\tilde{B}$, it follows that the minimal positive solution $\tilde{X}$ of (2.2) satisfies the following relations:

$$\Gamma - \phi\tilde{\xi}^T = \tilde{R}_1, \qquad \sigma(\tilde{R}_1) = \{\lambda_1, \ldots, \lambda_n\}, \tag{3.3}$$

$$\Delta - \phi\tilde{\eta}^T = -\tilde{R}_2, \qquad \sigma(-\tilde{R}_2) = \{\nu_1, \ldots, \nu_n\}, \tag{3.4}$$

$$\tilde{X}\Gamma + \Delta\tilde{X} = \tilde{\eta}\tilde{\xi}^T, \tag{3.5}$$

where

$$\tilde{\xi} = (I + \tilde{X}^T)\phi, \qquad \tilde{\eta} = (I + \tilde{X})\phi.$$

The last equation is a reformulation of (2.2). It thus follows that if the vectors $\tilde{\xi}$ and $\tilde{\eta}$ can be determined, then $\tilde{X}$ can be easily formulated based on the simple Sylvester equation (3.5).

The following result shows that $\tilde{\xi}$ and $\tilde{\eta}$ can be determined based on the relations (3.3) and (3.4).

PROPOSITION 3.2 (see [26]). A, B $A = \operatorname{diag}(a_1, \ldots, a_n)$ $a_1, \ldots, a_n \in \mathbb{R}$ $B \in \mathbb{R}^{n,n}$ $\lambda(B) = \{b_1, \ldots, b_n\}$ $b_1, \ldots, b_n \in \mathbb{R}$ $q_1, q_2, \ldots, q_n \in \mathbb{R} \setminus \{0\}$

$$q = [q_1, q_2, \ldots, q_n]^T, \quad Q = \operatorname{diag}(q_1, q_2, \ldots, q_n)$$

$$f = \left[\frac{\prod_{j=1}^{n}(a_1 - b_j)}{\prod_{j\neq 1}(a_1 - a_j)}, \ldots, \frac{\prod_{j=1}^{n}(a_k - b_j)}{\prod_{j\neq k}(a_k - a_j)}, \ldots, \frac{\prod_{j=1}^{n}(a_n - b_j)}{\prod_{j\neq n}(a_n - a_j)} \right]^T.$$

$z \in \mathbb{R}^n$ $A - qz^T = B$

$$z = Q^{-1} f = \left[\frac{f_1}{q_1}, \ldots, \frac{f_n}{q_n} \right]^T. \tag{3.6}$$

Using (3.6), (3.3), (3.4), and (3.5), we obtain the following explicit formulas for X.

THEOREM 3.3. (1.1). $k = 1, \ldots, n$

$$\xi_k = \frac{\prod_{j=1}^{n}(\gamma_k - \lambda_j)}{\prod_{j\neq k}(\gamma_k - \gamma_j)}, \quad \eta_k = \frac{\prod_{j=1}^{n}(\delta_k - \nu_j)}{\prod_{j\neq k}(\delta_k - \delta_j)}, \quad \kappa_k = \frac{\prod_{j=1}^{n}(\gamma_k + \delta_j)}{\prod_{j=1}^{n}(\gamma_k + \nu_j)}, \quad \epsilon_k = \frac{\prod_{j=1}^{n}(\delta_k + \gamma_j)}{\prod_{j=1}^{n}(\delta_k + \lambda_j)},$$

$$\begin{aligned} \xi &= [\xi_1, \ldots, \xi_n]^T, & \Xi &= \operatorname{diag}(\xi_1, \ldots, \xi_n), \\ \eta &= [\eta_1, \ldots, \eta_n]^T, & E &= \operatorname{diag}(\eta_1, \ldots, \eta_n), \\ \kappa &= [\kappa_1, \ldots, \kappa_n]^T, & K &= \operatorname{diag}(\kappa_1, \ldots, \kappa_n), \\ \epsilon &= [\epsilon_1, \ldots, \epsilon_n]^T, & \mathcal{E} &= \operatorname{diag}(\epsilon_1, \ldots, \epsilon_n), \end{aligned} \tag{3.7}$$

$$\Theta = \left[\frac{1}{\delta_i + \gamma_j} \right].$$

$$P = \operatorname{diag}(p_1, \dots, p_n),$$

p_i (1.1). (1.1):

$$X = P^{-1} E \Theta \Xi P^{-1}, \tag{3.8}$$

$$X = P^{-1} E \Theta K, \tag{3.9}$$

$$X = \mathcal{E} \Theta \Xi P^{-1}, \tag{3.10}$$

$$X = \mathcal{E} \Theta K. \tag{3.11}$$

. To prove the formulas, we apply Proposition 3.2 to (3.3) and obtain

$$\tilde{\xi} = \Phi^{-1} \xi,$$

where ξ is defined in (3.7). Similarly, from (3.4) we obtain

$$\tilde{\eta} = \Phi^{-1} \eta,$$

where η is defined in (3.7). By solving the Sylvester equation (3.4) we obtain

$$\tilde{X} = \Phi^{-1} E \Theta \Xi \Phi^{-1},$$

with E, Ξ as in (3.7). Then, (3.8) follows by using $X = \Phi^{-1} \tilde{X} \Phi^{-1}$ and $P = \Phi^2$.

In order to get the other formulas we need only show that $\Xi = PK$ and $E = P\mathcal{E}$.

Since $-\nu_1, \dots, -\nu_n, \lambda_1, \dots, \lambda_n$ are the eigenvalues of $\tilde{H}$, it follows from (2.5) that

$$\prod_{j=1}^{n} (\lambda - \lambda_j) \prod_{j=1}^{n} (\lambda + \nu_j) = \sum_{m=1}^{n} p_m \prod_{j \neq m} (\lambda - \gamma_j) \prod_{j=1}^{n} (\lambda + \delta_j)$$

$$- \sum_{m=1}^{n} p_m \prod_{j=1}^{n} (\lambda - \gamma_j) \prod_{j \neq m} (\lambda + \delta_j) + \prod_{j=1}^{n} (\lambda - \gamma_j) \prod_{j=1}^{n} (\lambda + \delta_j). \tag{3.12}$$

By inserting $\lambda = \gamma_k$, we obtain

$$\prod_{j=1}^{n} (\gamma_k - \lambda_j) \prod_{j=1}^{n} (\gamma_k + \nu_j) = p_k \prod_{j \neq k} (\gamma_k - \gamma_j) \prod_{j=1}^{n} (\gamma_k + \delta_j),$$

which implies that

$$\xi_k = p_k \kappa_k, \quad k = 1, 2, \dots, n.$$

We then have $\Xi = PK$.

Similarly, by inserting $\lambda = -\delta_k$ in (3.12) we get

$$\eta_k = p_k \epsilon_k, \quad k = 1, \dots, n,$$

and thus $E = P\mathcal{E}$. Then the other formulas follow. □

Note that formula (3.9) needs only the eigenvalues $-\nu_1, \dots, -\nu_n$, while formula (3.10) needs only the eigenvalues $\lambda_1, \dots, \lambda_n$. Numerically, these two formulas provide very cheap procedures to compute the minimal solution X of (1.1).

3.4. In [20] an explicit formula for the minimal solution of (1.1) was already given that is equivalent to (3.10). However, there a different expression for ϵ_k was introduced as

$$\epsilon_k = 1 + \sum_{m=1}^{n} \frac{1}{\delta_k + \lambda_m} \frac{\prod_{j=1}^{n}(\gamma_j - \lambda_m)}{\prod_{j\neq m}(\lambda_j - \lambda_m)}.$$

This expression is less compact and its evaluation has a higher complexity than the expression in Theorem 3.3.

In this section we have derived new explicit formulas for the minimal solution X of (1.1) and we will use them in the next section to derive some further properties of X.

4. Properties and bounds for the minimal positive solution. The simple expressions of the quantities $\xi_k, \kappa_k, \eta_k, \epsilon_k$ in the explicit formulas (3.8)–(3.11) and the eigenvalue interlacing property for the eigenvalues of $\tilde{H}$ allow one to derive further properties of the minimal positive solution of (1.1). For this we first prove the following lemma.

LEMMA 4.1. *ffi… γ_k, δ_k… (1.1),… $-\nu_k, \lambda_k$… $\tilde{H}$… (2.3),… $\xi_k, \eta_k, \kappa_k, \epsilon_k$, $k = 1, \ldots, n$… (3.7)…*

1.

$$0 < a_k < \eta_k < \delta_k - \nu_1 \le \delta_k, \qquad 0 < b_k < \xi_k < \gamma_k - \lambda_1 \le \gamma_k,$$

$$1 < \epsilon_k < \frac{\delta_k + \gamma_n}{\delta_k + \lambda_1} \le \frac{\delta_k + \gamma_n}{\delta_k}, \qquad 1 < \kappa_k < \frac{\gamma_k + \delta_n}{\gamma_k + \nu_1} \le \frac{\gamma_k + \delta_n}{\gamma_k},$$

…

$$a_k = \begin{cases} \frac{(\delta_k - \nu_k)(\nu_{k+1} - \delta_k)}{\delta_n - \delta_k}, & 1 \le k < n, \\ \delta_n - \nu_n, & k = n, \end{cases}$$

$$b_k = \begin{cases} \frac{(\gamma_k - \lambda_k)(\lambda_{k+1} - \gamma_k)}{\gamma_n - \gamma_k}, & 1 \le k < n, \\ \gamma_n - \lambda_n, & k = n. \end{cases}$$

2.

$$1 < \epsilon_n < \epsilon_{n-1} < \cdots < \epsilon_1, \qquad 1 < \kappa_n < \kappa_{n-1} < \cdots < \kappa_1.$$

…. To prove the first part, we use the interlacing property in Lemma 2.1 and obtain

$$0 < \frac{\delta_k - \nu_j}{\delta_k - \delta_{j-1}} < 1, \quad 1 < j \le k; \qquad \frac{\delta_k - \nu_j}{\delta_k - \delta_j} > 1, \quad 1 \le j < k,$$

and

$$0 < \frac{\delta_k - \nu_j}{\delta_k - \delta_j} < 1, \quad k < j \le n; \qquad \frac{\delta_k - \nu_{j+1}}{\delta_k - \delta_j} > 1, \quad k < j < n.$$

For $1 \le k < n$

$$\eta_k = \frac{(\delta_k - \nu_k)(\delta_k - \nu_{k+1})}{\delta_k - \delta_n} \prod_{j=1}^{k-1} \frac{\delta_k - \nu_j}{\delta_k - \delta_j} \prod_{j=k+1}^{n-1} \frac{\delta_k - \nu_{j+1}}{\delta_k - \delta_j} > a_k,$$

and

$$\eta_k = (\delta_k - \nu_1) \prod_{j=1}^{k-1} \frac{\delta_k - \nu_{j+1}}{\delta_k - \delta_j} \prod_{j=k+1}^{n} \frac{\delta_k - \nu_j}{\delta_k - \delta_j} < \delta_k - \nu_1 \le \delta_k.$$

Finally, for $k = n$ we obtain

$$\eta_n = (\delta_n - \nu_n) \prod_{j=1}^{n-1} \frac{\delta_n - \nu_j}{\delta_n - \delta_j} > \delta_n - \nu_n =: a_n$$

and

$$\eta_n = (\delta_n - \nu_1) \prod_{j=1}^{n-1} \frac{\delta_n - \nu_{j+1}}{\delta_n - \delta_j} < \delta_n - \nu_1 \le \delta_n.$$

This proves the inequalities for the η_k, and clearly we have $a_k > 0$ for $k = 1, \ldots, n$.

The inequalities for the ξ_k can be derived in the same way by using the interlacing property for the eigenvalues $\lambda_1, \ldots, \lambda_n$. This interlacing property also gives

$$\epsilon_k = \prod_{j=1}^{n} \frac{\delta_k + \gamma_j}{\delta_k + \lambda_j} > 1$$

and

$$\epsilon_k = \frac{\delta_k + \gamma_n}{\delta_k + \lambda_1} \prod_{j=1}^{n-1} \frac{\delta_k + \gamma_j}{\delta_k + \lambda_{j+1}} < \frac{\delta_k + \gamma_n}{\delta_k + \lambda_1} \le \frac{\delta_k + \gamma_n}{\delta_k}.$$

Similarly, one can prove the inequalities for κ_k.

To prove part 2 we consider the function

$$\psi(t) = \prod_{j=1}^{n} \frac{t + \gamma_j}{t + \lambda_j} = \prod_{j=1}^{n} \left(1 + \frac{\gamma_j - \lambda_j}{t + \lambda_j}\right).$$

Since $\gamma_j - \lambda_j \ge 0$ for $j = 1, \ldots, n$, it follows that $\psi(t)$ is decreasing as t increases. Since $\psi(\delta_k) = \epsilon_k$ for $k = 1, \ldots, n$, and $\delta_1 < \cdots < \delta_n$, we thus have

$$\epsilon_1 > \epsilon_2 > \cdots > \epsilon_n.$$

Obviously $\psi(t) > 1$ for any $t > 0$, and hence $\epsilon_n = \psi(\delta_n) > 1$.

The monotonicity $\kappa_1 > \cdots > \kappa_n > 1$ follows in the same way. □

With the help of Lemma 4.1 we can now prove the following entrywise monotonicity property of the minimal positive solution X of (1.1).

THEOREM 4.2. [illegible] $X = [x_{ij}] \in \mathbb{R}^{n,n}$ [illegible] (1.1). [illegible] $i \ge k$ [illegible] $j \ge l$ [illegible] $(i,j) \ne (k,l)$ [illegible] X [illegible]

$$x_{ij} > x_{kl}.$$

[illegible]. Since

$$0 < \gamma_1 < \cdots < \gamma_n, \quad 0 < \delta_1 < \cdots < \delta_n,$$

and by Lemma 4.1,

$$1 < \epsilon_n < \cdots < \epsilon_1, \quad 1 < \kappa_n < \cdots < \kappa_1,$$

with (3.11), for $1 \le i, j \le n$, if $i < n$, it follows that

$$x_{ij} = \frac{\epsilon_i \kappa_j}{\delta_i + \gamma_j} > \frac{\epsilon_{i+1} \kappa_j}{\delta_{i+1} + \gamma_j} = x_{i+1,j}.$$

If $j < n$, then

$$x_{ij} = \frac{\epsilon_i \kappa_j}{\delta_i + \gamma_j} > \frac{\epsilon_i \kappa_{j+1}}{\delta_i + \gamma_{j+1}} = x_{i,j+1}. \qquad \square$$

The quantities in Lemma 4.1 also provide upper and lower bounds for the entries of the minimal positive solution X of (1.1).

THEOREM 4.3. [illegible] $X = [x_{ij}] \in \mathbb{R}^{n,n}$ [illegible] (1.1).

$$\frac{w_{ij}}{\delta_i + \gamma_j} < x_{ij} < \frac{W_{ij}}{\delta_i + \gamma_j},$$

[illegible]

$$w_{ij} = \max\left\{ \frac{a_i b_j}{p_i p_j},\ \frac{a_i}{p_i},\ \frac{b_j}{p_j},\ 1 \right\},$$

$$W_{ij} = \min\left\{ \frac{\delta_i \gamma_j}{p_i p_j},\ \frac{\delta_i(\gamma_j + \delta_n)}{p_i \gamma_j},\ \frac{(\delta_i + \gamma_n)\gamma_j}{\delta_i p_j},\ \frac{(\delta_i + \gamma_n)(\gamma_j + \delta_n)}{\delta_i \gamma_j} \right\}.$$

[illegible]. The bounds follow from the formulas (3.8)–(3.11) and the inequalities given in the first part of Lemma 4.1. $\square$

COROLLARY 4.4. [illegible] $X = [x_{ij}] \in \mathbb{R}^{n,n}$ [illegible] (1.1) [illegible] w_{ij}, W_{ij} [illegible] 4.3. [illegible]

$$\frac{w_{nn}}{\delta_n + \gamma_n} < x_{nn} \le x_{ij} \le x_{11} < \frac{W_{11}}{\delta_1 + \gamma_1}$$

[illegible] $i, j = 1, \ldots, n$

[illegible]. The inequalities follow from Theorems 4.2 and 4.3. $\square$

By taking advantage of the scaled equation (2.2), we also obtain a bound for the spectral norm of the minimal positive solution X of (1.1).

THEOREM 4.5. [illegible] $\tilde{X} \in \mathbb{R}^{n,n}$ [illegible] (2.2). [illegible]

$$\|\tilde{X}\| \le 1,$$

[illegible] $\|\tilde{X}\| = 1$, [illegible] $\chi(0) = 0$ [illegible] $\chi'(0) = 0$ [illegible] X [illegible] (1.1) [illegible]

$$\|X\| \le \frac{1}{\min_j p_j}.$$

[illegible]. Let $\tilde{X}_+ \ge \tilde{X}$ be another positive solution of (2.2) [20]. Since both $\tilde{X}$ and $\tilde{X}_+$ are positive, it is easily verified that

$$\|\tilde{X}\|^2 = \rho(\tilde{X}^T \tilde{X}) \le \rho(\tilde{X}^T \tilde{X}_+),$$

where $\rho(Z)$ is the spectral radius of Z. Lemma 3.1 shows that $\tilde{X}^T$ is the minimal positive solution of the dual equation (3.1). By Lemma 12 of [7], $\rho(\tilde{X}^T\tilde{X}_+) \le 1$. Hence $\|\tilde{X}\| \le 1$, and $\|\tilde{X}\| = 1$ if and only if $\tilde{X}_+ = \tilde{X}$. The last equality holds if and only if 0 is a double eigenvalue of $\tilde{H}$, which is equivalent to the conditions $\chi(0) = 0$ and $\chi'(0) = 0$, by Lemma 2.1.

The upper bound for $\|X\|$ follows from the relation $X = \Phi^{-1}\tilde{X}\Phi^{-1}$. □

Various lower bounds for $\|X\|$ can also be derived by using the inequalities for the entries of X, but we will not pursue this topic here.

At the end of this section we also provide a formula for the inverse of X.

THEOREM 4.6. [illegible] $X = [x_{ij}]$ [illegible] (1.1) [illegible] P, Θ [illegible] 3.3, [illegible]

$$X^{-1} = PQ\Theta^T GP,$$

[illegible]

$$Q = \operatorname{diag}(q_1, \dots, q_n), \qquad G = \operatorname{diag}(g_1, \dots, g_n),$$

[illegible]

$$q_k = \prod_{j=1}^{n} \frac{\gamma_k + \delta_j}{\gamma_k - \lambda_j}, \qquad g_k = \prod_{j=1}^{n} \frac{\delta_k + \gamma_j}{\delta_k - \nu_j},$$

[illegible] $k = 1, \dots, n$ [illegible]. Since $\gamma_n > \cdots > \gamma_1 > 0$ and $\delta_n > \cdots > \delta_1 > 0$, it follows (see, e.g., [6]) that the Cauchy matrix Θ is invertible and

$$\Theta^{-1} = \hat{Q}\Theta^T\hat{G},$$

where

$$\hat{Q} = \operatorname{diag}(\hat{q}_1, \dots, \hat{q}_n), \quad \hat{G} = \operatorname{diag}(\hat{g}_1, \dots, \hat{g}_n),$$

with

$$\hat{q}_k = \frac{\displaystyle\prod_{j=1}^{n}(\gamma_k + \delta_j)}{\displaystyle\prod_{j\ne k}(\gamma_k - \gamma_j)}, \qquad \hat{g}_k = \frac{\displaystyle\prod_{j=1}^{n}(\delta_k + \gamma_j)}{\displaystyle\prod_{j\ne k}(\delta_k - \delta_j)},$$

for $k = 1, \dots, n$. Since all of the diagonal matrices in (3.8) are invertible, it follows that X is also invertible and the formula for X^{-1} follows from (3.8) using Θ^{-1}. □

5. Numerical algorithms. The formulas given in section 3 can be used to develop the following numerical algorithms for computing the minimal positive solution of (1.1).

ALGORITHM 5.1. For the Riccati equation (1.1) this algorithm computes the minimal positive solution.

1. Compute the eigenvalues $-\nu_1, \dots, -\nu_n, \lambda_1, \dots, \lambda_n$ of $\tilde{H}$ in (2.3) by applying a root finding solver to the secular equation $\chi(\lambda) = 0$ given by (2.4).
2. Use either of the formulas (3.8) or (3.11) to compute the minimal positive solution X of (1.1).

We can also use either of the formulas (3.9) or (3.10).

ALGORITHM 5.2. For the Riccati equation (1.1) this algorithm computes the minimal positive solution.

1. Compute the eigenvalues $-\nu_1, \ldots, -\nu_n$ of $\tilde{H}$ in (2.3) by applying a root finding solver to the secular equation $\chi(\lambda) = 0$ given by (2.4).
2. Use formula (3.9) to compute the minimal positive solution X of (1.1).

ALGORITHM 5.3. For the Riccati equation (1.1) this algorithm computes the minimal positive solution.

1. Compute the eigenvalues $\lambda_1, \ldots, \lambda_n$ of $\tilde{H}$ in (2.3) by applying a secular equation solver to $\chi(\lambda) = 0$.
2. Use formula (3.10) to compute the minimal positive solution X of (1.1).

Note that Algorithms 5.2 and 5.3 need only compute half of the eigenvalues.

The success of these three algorithms depends on how fast and accurately the eigenvalues can be computed and how sensitive the evaluation of the formulas (3.8)–(3.11) is. This requires an efficient and reliable secular equation solver. The osculatory interpolation methods of [3, 24] that were developed in the context of the divide-and-conquer eigenvalue methods ([10, section 8.5], [4, 5, 9]) may not be applicable directly, since the secular function $\chi(\lambda)$ has quite different properties than the secular equation derived in the symmetric divide-and-conquer method. For this reason we propose the following hybrid method for the computation of roots of the secular function. We consider only the case for computing the eigenvalues λ_k as the method for computing the eigenvalues ν_k is analogous. Our approach treats λ_1 differently from the other eigenvalues $\lambda_2, \ldots, \lambda_n$, because of the different properties that λ_1 has.

5.1. Computation of λ_k with $k > 1$.

1. Initial guess. To compute an initial guess, we basically follow the procedure suggested in [24]. We first evaluate $\chi(m_k)$, where m_k is the midpoint of the interval (γ_{k-1}, γ_k). Because $\chi(\lambda)$ has only one root in (γ_{k-1}, γ_k), and since $\lim_{\lambda \to \gamma_{k-1}^+} \chi(\lambda) = \infty$, and $\lim_{\lambda \to \gamma_k^-} \chi(\lambda) = -\infty$, based on the sign of $\chi(m_k)$, we can easily determine in which half of the interval λ_k is located. Simple geometry shows that if $\chi(m_k) > 0$, then λ_k is closer to γ_k, and if $\chi(m_k) < 0$, then λ_k is closer to γ_{k-1}. We then consider the equation

$$\frac{p_{k-1}}{\lambda - \gamma_{k-1}} + \frac{p_k}{\lambda - \gamma_k} + r_k = 0,$$

with $r_k = \chi(m_k) - p_{k-1}/(m_k - \gamma_{k-1}) - p_k/(m_k - \gamma_k)$, which can be obtained during the evaluation of $\chi(m_k)$ without any extra cost. We then take the root of this equation in (γ_{k-1}, γ_k) as our initial guess z_k^0. It is easily verified that z_k^0 and λ_k are in the same half interval. We also choose an initial interval so that the χ values on endpoints have opposite signs (which guarantees that λ_k is in this interval). If $\chi(m_k)\chi(z_k^0) < 0$, then we use m_k, z_k^0 for the interval. Otherwise, we use the asymptotic properties of χ to find another λ value to replace m_k. Let us denote the resulting interval by $[u_0, v_0]$.

2. Iteration step. For a current approximation z_k^j, we first evaluate $\chi'(z_k^j)$ and use one step of Newton's method to determine the next approximate z_k^{j+1}. If z_k^{j+1} is inside the current interval $[u_j, v_j]$, then we evaluate $\chi(z_k^{j+1})$. We then replace one of u_j, v_j and its corresponding χ value with z_k^{j+1} and $\chi(z_k^{j+1})$ based on the sign of $\chi(z_k^{j+1})$ and move on to the next iteration. If z_k^{j+1} is outside $[u_j, v_j]$ (maybe even outside of (γ_{k-1}, γ_k)), then we apply one step of

the secant method with u_j, v_j and their corresponding χ values to get z_k^{j+1}. We then evaluate $\chi(z_k^{j+1})$, update $[u_j, v_j]$, and continue. If this z_k^{j+1} is still outside of $[u_j, v_j]$, then we use one step of the bisection method with u_j, v_j to get z_k^{j+1}.

When the iterates z_k^j get close to the root λ_j, then, due to rounding errors, it becomes more difficult to compute a reliable value of $\chi(z_k^j)$. (This happens typically for small roots.) This may cause the sign of χ to alternate between positive and negative values in the Newton iteration and the secant iteration, which may have the effect that the sequence $\{z_k^j\}$ does not converge. If we observe such a behavior and the function values for χ are also small in absolute value, then we run a step of the bisection method. This procedure has turned out to be very successful during our numerical tests.

3. Stopping criterion. In order to compute the root λ_k accurately, we actually use the shift $s = \lambda - \gamma_{k-1}$ or $s = \lambda - \gamma_k$ initially, depending on whether λ_k is closer to γ_{k-1} or γ_k. The iteration step is then applied to the new variable s to generate a sequence of approximate values $s_0, s_1, \ldots, s_j, \ldots$. The iteration can be written as

$$s_{j+1} = s_j + \Delta s_j,$$

where Δs_j is the jth correction.

We use the stopping criterion

$$|\Delta s_j| < c\varepsilon_M |s_{j+1}|, \tag{5.1}$$

where ε_M is the machine epsilon and c is a modest constant (which is set to 48 in our tests).

The procedure for the computation of ν_k $(k = 2, \ldots, n)$ is analogous.

5.2. Computation of λ_1.

1. Initial guess. The strategy for choosing starting values z_1^0 and starting intervals $[u_0, v_0]$ is slightly different than in the case of the other eigenvalues. Since we know that $\lambda_1 \in [0, \gamma_1)$, we first evaluate $\chi(m_1)$, where $m_1 = \gamma_1/2$. We use the sign of $\chi(m_1)$ to determine if λ_1 is closer to 0 or γ_1. We then use the root $z_1^0 \in [0, \gamma)$ of the equation

$$\frac{p_1}{\lambda - \gamma_1} + r_1 = 0,$$

with $r_1 = \chi(m_1) - p_1/(m_1 - \gamma_1)$, as the initial starting value.

If $\chi(m_1), \chi(z_1^0) < 0$, then we use m_1, z_1^0 to form the initial interval $[u_0, v_0]$. If $\chi(m_1), \chi(z_1^0) > 0$, then we replace m_1 by another value such that the corresponding χ value is negative, by using the fact $\lim_{\lambda \to \gamma_1^-}(\lambda) = -\infty$. In the case that $\chi(m_1), \chi(z_1^0) < 0$, if $\chi(0) > 0$, we replace m_1 with 0. If $\chi(0) = 0$, we still need to check the sign of $\chi'(0)$. If $\chi'(0) > 0$, we may use it to find a small positive number such that its corresponding χ is positive. We then replace m_1 with this number. If $\chi'(0) \leq 0$, we simply set $\lambda_1 = 0$, and no iteration is required.

Note that for the transport theory problem, $\chi(0)$ and $\chi'(0)$ can be easily determined by the formulas given in Remark 2.2.

2. Iteration step. We first use the same iteration steps as described for the eigenvalues λ_k, $k \geq 2$, to an approximation of λ_1. This usually works well for $\lambda_1 > c_1\sqrt{\varepsilon_M}$ with some positive constant c_1. If, however, λ_1 is too small, then it is difficult to get accurate function values for χ and χ', which then may cause convergence problems. In order to overcome this difficulty, once we observe that the jth approximate z_1^j satisfies $z_1^j < c_1\sqrt{\varepsilon_M}$ (we used $c_1 = 100$ in our tests), we evaluate $\chi(z_1^j)$ and $\chi'(z_1^j)$ by using their corresponding Taylor polynomials at 0, given by

$$\chi(z_1^j) \approx \chi(0) + z_1^j\chi'(0) + \frac{(z_1^j)^2}{2}\chi''(0),$$

$$\chi'(z_1^j) \approx \chi'(0) + z_1^j\chi''(0) + \frac{(z_1^j)^2}{2}\chi'''(0),$$

and use these values in the next step of the Newton iteration. If $\chi'(z_1^j)$ is also very small in modulus, then we approximate $\chi''(z_1^j)$ by

$$\chi''(z_1^j) \approx \chi''(0) + z_1^j\chi'''(0).$$

We then use the approximations for $\chi(z_1^j)$, $\chi'(z_1^j)$, $\chi''(z_1^j)$ to construct the second degree Taylor polynomial for χ at z_1^j and use one of the roots of this polynomial (if it exists) as our next iterate z_1^{j+1}.
For a general secular equation, the computation of $\chi(0)$, $\chi'(0)$, $\chi''(0)$, and $\chi'''(0)$ requires extra cost and it is not clear if the values can be evaluated accurately. In the secular equation from the transport problem, however, this computation is essentially cost-free since we may use the formulas in Remark 2.2, and because of the simple formulations the values can be computed accurately.

3. Stopping criterion. We use again the stopping criterion (5.1) (with $\gamma_0 := 0$).

The procedure for the computation of ν_1 is analogous.

5.3. Costs. The main cost in Algorithms 5.1–5.3 is the evaluation of χ and χ' during each iteration step. In order to evaluate $\chi(\lambda)$ and $\chi'(\lambda)$, we first compute $\lambda - \gamma_j$, $\lambda + \delta_j$ for $j = 1, \ldots, n$. We then compute $p_j/(\lambda - \gamma_j)$ and $p_j/(\lambda + \delta_j)$. After this $\chi(\lambda)$ can be evaluated. We continue to compute $[p_j/(\lambda - \gamma_j)]/(\lambda - \gamma_j)$ and $[p_j/(\lambda+\delta_j)]/(\lambda+\delta_j)]$, which costs one extra flop for each term, and then evaluate $\chi'(\lambda)$. So if the Newton iteration is used in the iteration step, then the cost per iteration step and per eigenvalue is about $10n$ flops. If the average number of iterations is M, then the cost for Algorithm 5.1 is about $(20M + 9)n^2$ flops, and the cost for Algorithms 5.2 and 5.3 is about $(10M + 9)n^2$ flops. Note that it requires $3n^2$ flops to compute each set of the values $\xi_k, \eta_k, \kappa_k, \epsilon_k$, and it requires another $3n^2$ flops to compute the components of X. Note also that in these complexity estimates we did not count the cost for the computation of the initial values.

5.4. Error analysis. To analyze the computational errors in the described procedures, we first estimate the errors in the computed eigenvalues; see also [30]. We assume that the iteration for each eigenvalue stops when (5.1) holds, and the computed sequence satisfies the conditions in the following lemma observed by Kahan (see, e.g., [24]).

LEMMA 5.4. $\{x_j\}_{j=1}^{\infty}$ $\lim_{j\to\infty} x_j = x^*$ $\frac{|x_{j+1}-x_j|}{|x_j-x_{j-1}|}$ $j \geq k$, $\frac{|x_{k+1}-x_k|}{|x_k-x_{k-1}|} < 1$

$$|x_{k+1} - x^*| < \frac{|x_{k+1} - x_k|^2}{|x_k - x_{k-1}| - |x_{k+1} - x_k|}.$$

Let λ_j, ν_j be the exact eigenvalues of H, and let $\hat{\lambda}_j$, $\hat{\nu}_j$ be the corresponding computed eigenvalues. With the discussed properties of the eigenvalues, the presented procedures, and Lemma 5.4, it is reasonable to assume that the computed eigenvalues satisfy

$$|\lambda_j - \hat{\lambda}_j| < C_{\lambda_j}\varepsilon_M \min\{\gamma_j - \lambda_j,\, \lambda_j - \gamma_{j-1}\}, \tag{5.2}$$

$$|\nu_j - \hat{\nu}_j| < C_{\nu_j}\varepsilon_M \min\{\delta_j - \nu_j,\, \nu_j - \delta_{j-1}\}, \tag{5.3}$$

for $j = 1, \ldots, n$, where $\gamma_0 = \delta_0 = 0$ and C_{λ_j}, C_{ν_j} are some modest constants. We then have the following lemma.

LEMMA 5.5. $\hat{\lambda}_j$, $-\hat{\nu}_j$, H (1.5) (5.2) (5.3). $\hat{\xi}_k$, $\hat{\eta}_k$, $\hat{\epsilon}_k$, $\hat{\kappa}_k$ 3.3.

$$\hat{\xi}_k = \xi_k(1 + nC_{\xi_k}\varepsilon_M), \quad \hat{\eta}_k = \eta_k(1 + nC_{\eta_k}\varepsilon_M),$$

$$\hat{\kappa}_k = \kappa_k(1 + nC_{\kappa_k}\varepsilon_M), \quad \hat{\epsilon}_k = \epsilon_k(1 + nC_{\epsilon_k}\varepsilon_M),$$

$k = 1, \ldots, n$, C_{ξ_k}, C_{η_k}, C_{κ_k}, C_{ϵ_k} . For the proof we just consider the first order error.

Note that $\hat{\xi}_k$ is actually computed by the formula

$$\prod_{j=1}^{n}(\gamma_k - \hat{\lambda}_j) / \prod_{j\neq k}(\gamma_k - \gamma_j);$$

i.e., λ_j is replaced with $\hat{\lambda}_j$. We then have

$$|\gamma_k - \hat{\lambda}_j| = |(\gamma_k - \lambda_j) + (\lambda_j - \hat{\lambda}_j)| = |\gamma_k - \lambda_j|\left|1 + \frac{\lambda_j - \hat{\lambda}_j}{\gamma_k - \lambda_j}\right| =: |\gamma_k - \lambda_j||1 + \tilde{C}_{kj}\varepsilon_M|,$$

for $j = 1, \ldots, n$, where by (5.2) and the interlacing property of the eigenvalues

$$|\tilde{C}_{kj}| = \frac{1}{\varepsilon_M}\left|\frac{\lambda_j - \hat{\lambda}_j}{\gamma_k - \lambda_j}\right| < C_{kj}\frac{\min\{\gamma_j - \lambda_j,\, \lambda_j - \gamma_{j-1}\}}{|\gamma_k - \lambda_j|} \leq C_{kj}.$$

With this relation, it is not difficult to obtain that

$$\hat{\xi}_k = \xi_k(1 + nC_{\xi_k}\varepsilon_M),$$

where C_{ξ_k} is a constant. The corresponding relations for the other terms follow in the same way. □

Using this lemma we obtain the following relative errors for the components of the minimal positive solution computed by the formulas given in section 3.

THEOREM 5.6. $X = [x_{ij}]$ (1.1) (3.8)–(3.11) (5.2) (5.3) $\hat{X} = [\hat{x}_{ij}]$

$$\frac{|\hat{x}_{ij} - x_{ij}|}{x_{ij}} = D_{ij} n \varepsilon_M, \qquad i, j = 1, \ldots, n$$

D_{ij} . The relative error estimates follow from Lemma 5.5. □

6. Numerical examples. In this section we present some numerical test results for the problems from transport theory; see [20, 21]. The weights $c_1, \ldots, c_n$ and nodes $\omega_1, \ldots, \omega_n$ are generated from the composite four-node Gauß–Legendre quadrature formula on [0, 1] with $n/4$ equally spaced subintervals; see, e.g., [29]. All the numerical examples were tested in MATLAB version 7.1.0 with machine precision $\varepsilon_M \approx 2.22e-16$. We solved the problem for various numbers of the parameters α and β and the size n. We used all four formulas to compute the minimal positive solution, with a secular equation solver as described in section 5.

The computed minimal positive solutions via formulas (3.8)–(3.11) are denoted by $X^{(1)}, X^{(2)}, X^{(3)}, X^{(4)}$, respectively. In the following we display the test results. We present one table for each pair (α, β) and various values of n. (The used norm is always the spectral norm.) In each of Tables 6.1–6.6, we list the following results:

- Maximum residual:

$$R = \max_{j \in \{1,2,3,4\}} \|X^{(j)}\Gamma + \Delta X^{(j)} - (e + X^{(j)}p)(e^T + p^T X^{(j)})\|.$$

- Maximum and minimum entrywise relative errors:

$$RE_{\max} = \max_{\substack{i,j \in \{1,2,3,4\} \\ i \neq j}} \max_{k,l \in \{1,\ldots,n\}} \frac{|x_{kl}^{(i)} - x_{kl}^{(j)}|}{\min\{x_{kl}^{(i)}, x_{kl}^{(j)}\}},$$

$$RE_{\min} = \min_{\substack{i,j \in \{1,2,3,4\} \\ i \neq j}} \max_{k,l \in \{1,\ldots,n\}} \frac{|x_{kl}^{(i)} - x_{kl}^{(j)}|}{\min\{x_{kl}^{(i)}, x_{kl}^{(j)}\}}.$$

- Largest entry x_{11} (determined by one of the four solutions).
- Smallest entry x_{nn} (determined by one of the four solutions).
- Norm $\|X\|$ (X is one of the four solutions). Note that we have proved that $\|\tilde{X}\| \leq 1$, which translates to $\|X\| \leq 1/\min p_j$.
- Number of iterations for ν_1: N_-.
- Number of iterations for λ_1: N_+.
- Average of the number of iterations for all $2n$ eigenvalues: N.

We also give the eigenvalues $-\nu_1, \lambda_1$ in the caption.

We can summarize the numerical results as follows.

1. The values of R in the tables are usually the residual of $X^{(1)}$. The other residuals are basically the same, but some can be one order smaller.
2. Since we do not know the exact solution, we use $RE_{\max}$ and $RE_{\min}$ to detect if high relative accuracy can actually be achieved. The values of $RE_{\max}$ and $RE_{\min}$ do support the high relative accuracy result. (Note that x_{nn} is small in all examples.)

3. The number of iterations for ν_1 and λ_1 increases as $\alpha \to 0$ and $\beta \to 1$. This shows the numerical difficulty when the eigenvalues $-\nu_1$ and λ_1 are getting close to each other. However, our computed values of ν_1, λ_1 are much more accurate than those obtained by running the MATLAB code .. on $\tilde{H}$.
4. Our MATLAB implementation of the root finder based on the secular equation is still not very robust. In general, about .5% of the eigenvalues need 100 iterations, the maximum iteration number used in our experimental code. Some further improvement could enhance these convergence properties.

TABLE 6.1
$\alpha = 0.5,\ \beta = 0.5,\ \ (-\nu_1, \lambda_1) \approx (-1.166, 3.996)$.

n	R	$RE_{\max}$	$RE_{\min}$	x_{11}	x_{nn}	$\|X\|$	N_-	N_+	N
64	2.70e-13	1.83e-14	6.80e-15	.263	8.23e-04	7.87e+00	8	7	5
128	1.27e-12	6.72e-14	3.33e-14	.263	4.09e-04	1.57e+01	9	8	5
256	5.35e-12	1.64e-13	7.73e-14	.264	2.04e-04	3.15e+01	9	9	5
512	1.97e-11	2.70e-13	1.34e-13	.264	1.02e-04	6.29e+01	10	8	5

TABLE 6.2
$\alpha = 0.1,\ \beta = 0.99,\ \ (-\nu_1, \lambda_1) \approx (-7.98e-02, 3.83e-01)$.

n	R	$RE_{\max}$	$RE_{\min}$	x_{11}	x_{nn}	$\|X\|$	N_-	N_+	N
64	5.16e-13	2.65e-14	1.23e-14	2.70	2.19e-03	6.12e+01	8	6	5
128	2.43e-12	9.67e-14	4.06e-14	2.72	1.08e-03	1.22e+02	10	5	5
256	8.48e-12	1.46e-13	7.03e-14	2.72	5.37e-04	2.45e+02	9	5	5
512	3.48e-11	4.21e-13	2.04e-13	2.72	2.67e-04	4.89e+02	10	6	6

TABLE 6.3
$\alpha = 10^{-4},\ \beta = 1 - 10^{-8},\ \ (-\nu_1, \lambda_1) \approx (-7.91e-05, 3.79e-04)$.

n	R	$RE_{\max}$	$RE_{\min}$	x_{11}	x_{nn}	$\|X\|$	N_-	N_+	N
64	2.46e-11	1.48e-12	7.35e-13	4.19	2.24e-03	8.59e+01	23	16	5
128	1.02e-10	5.16e-12	2.57e-12	4.21	1.10e-03	1.72e+02	26	25	5
256	4.66e-11	1.24e-12	5.60e-13	4.22	5.48e-04	3.43e+02	19	25	5
512	5.43e-10	7.02e-12	3.48e-12	4.22	2.73e-04	6.87e+02	34	25	6

TABLE 6.4
$\alpha = 10^{-14},\ \beta = 1 - 10^{-14},\ \ (-\nu_1, \lambda_1) \approx (-1.73e-07, 1.73e-07)$.

n	R	$RE_{\max}$	$RE_{\min}$	x_{11}	x_{nn}	$\|X\|$	N_-	N_+	N
64	6.09e-13	2.52e-14	1.02e-14	4.19	2.24e-03	8.59e+01	28	26	6
128	2.72e-12	7.80e-14	3.15e-14	4.21	1.10e-03	1.72e+02	28	26	5
256	1.02e-11	1.85e-13	8.30e-14	4.22	5.48e-04	3.44e+02	28	26	5
512	4.28e-11	4.12e-13	1.60e-13	4.22	2.73e-04	6.87e+02	28	26	6

TABLE 6.5
$\alpha = 10^{-8},\ \beta = 1,\ \ (-\nu_1, \lambda_1) = (0, 3.00e-08)$.

n	R	$RE_{\max}$	$RE_{\min}$	x_{11}	x_{nn}	$\|X\|$	N_-	N_+	N
64	7.74e-13	4.84e-14	1.94e-14	4.19	2.24e-03	8.59e+01	0	30	5
128	2.95e-12	8.97e-14	4.07e-14	4.21	1.10e-03	1.72e+02	0	30	5
256	1.21e-11	1.76e-13	7.39e-14	4.22	5.48e-04	3.44e+02	0	32	5
512	4.51e-11	4.14e-13	1.87e-13	4.22	2.73e-04	6.87e+02	0	30	6

TABLE 6.6
$\alpha = 10^{-15},\ \beta = 1,\ \ (-\nu_1, \lambda_1) = (0, 3.00e-15).$

n	R	$RE_{\max}$	$RE_{\min}$	x_{11}	x_{nn}	$\|X\|$	N_-	N_+	N
64	6.97e-13	3.39e-14	1.42e-14	4.19	2.24e-03	8.59e+01	0	55	5
128	2.71e-12	7.83e-14	2.91e-14	4.21	1.10e-03	1.72e+02	0	55	5
256	1.02e-11	1.60e-13	7.47e-14	4.22	5.48e-04	3.44e+02	0	55	5
512	4.19e-11	3.71e-13	1.53e-13	4.22	2.73e-04	6.87e+02	0	55	5

7. Conclusion. We have presented four formulas for the minimal positive solution of the nonsymmetric Riccati equation (1.1) that depend on the eigenvalues of the associated matrix. With the help of the formulas we have given some properties and entrywise bounds for the minimal positive solution. We have used the formulas to develop fast numerical algorithms for computing the minimal positive solution. If the eigenvalues can be computed accurately, then the computed minimal positive solution has high relative accuracy.

Acknowledgments. We thank an anonymous referee for suggestions that helped to improve this paper. Hongguo Xu wishes to gratefully acknowledge the hospitality of TU Berlin, where part of this research was carried out.

REFERENCES

[1] Z.-Z. Bai, X.-X. Guo, and S.-F. Xu, *Alternately linearized implicit iteration methods for the minimal nonnegative solutions of the nonsymmetric algebraic Riccati equations*, Numer. Linear Algebra Appl., 13 (2006), pp. 655–674.
[2] D. A. Bini, B. Iannazzo, and F. Poloni, *A fast Newton's method for a nonsymmetric algebraic Riccati equation*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 276–290.
[3] J. R. Bunch, C. P. Nielson, and D. C. Sorensen, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.
[4] J. J. M. Cuppen, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1980/81), pp. 177–195.
[5] J. J. Dongarra and D. C. Sorensen, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. S139–S154.
[6] T. Finck, G. Heinig, and K. Rost, *An inversion formula and fast algorithms for Cauchy-Vandermonde matrices*, Linear Algebra Appl., 183 (1993), pp. 179–191.
[7] S. Fital and C.-H. Guo, *Convergence of the solution of a nonsymmetric matrix Riccati differential equation to its stable equilibrium solution*, J. Math. Anal. Appl., 318 (2006), pp. 648–657.
[8] B. D. Ganapol, *An investigation of a simple transport model*, Transport Theory Statist. Phys., 21 (1992), pp. 1–37.
[9] G. H. Golub, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–344.
[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
[11] M. Gu and S. C. Eisenstat, *A divide-and conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 172–191.
[12] C.-H. Guo, *Nonsymmetric algebraic Riccati equations and Wiener-Hopf factorization for M-matrices*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 225–242.
[13] C.-H. Guo, *A note on the minimal nonnegative solution of a nonsymmetric algebraic Riccati equation*, Linear Algebra Appl., 357 (2002), pp. 299–302.
[14] C.-H. Guo and N. J. Higham, *Iterative solution of a nonsymmetric algebraic Riccati equation*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 396–412.
[15] C.-H. Guo, B. Iannazzo, and B. Meini, *On the doubling algorithm for a (shifted) nonsymmetric algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1083–1100.
[16] C.-H. Guo and A. J. Laub, *On the iterative solution of a class of nonsymmetric algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 376–391.
[17] X.-X. Guo, W.-W. Lin, and S.-F. Xu, *A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equation*, Numer. Math., 103 (2006), pp. 393–412.

[18] J. Juang, *Existence of algebraic matrix Riccati equations arising in transport theory*, Linear Algebra Appl., 230 (1995), pp. 89–100.
[19] J. Juang and I.-D. Chen, *Iterative solution for a certain class of algebraic matrix Riccati equations arising in transport theory*, Transport Theory Statist. Phys., 22 (1993), pp. 65–80.
[20] J. Juang and W.-W. Lin, *Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 228–243.
[21] J. Juang and Z. T. Lin, *Convergence of an iterative technique for algebraic matrix Riccati equations and applications to transport theory*, Transport Theory Statist. Phys., 21 (1992), pp. 87–100.
[22] J. Juang, C. L. Hsing, and P. Nelson, *Global existence, asymptotics and uniqueness for the reflection kernel of the angularly shifted transport equation*, Math. Models Methods Appl. Sci., 5 (1995), pp. 239–251.
[23] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Oxford University Press, New York, 1995.
[24] R.-C. Li, *Solving Secular Equations Stably and Efficiently*, Technical Report UCB//CSD-94-851, LAPACK Working Note 93, Computer Science Division, Department of EECS, University of California, Berkeley, CA, 1994.
[25] L.-Z. Lu, *Solution form and simple iteration of a nonsymmetric algebraic Riccati equation arising in transport theory*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 679–685.
[26] V. Mehrmann and H. Xu, *Choosing poles so that the single-input pole placement problem is well conditioned*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 664–681.
[27] A. Melman, *Numerical solution of a secular equation*, Numer. Math., 69 (1995), pp. 483–493.
[28] L. C. G. Rogers, *Fluid models in queueing theory and Wiener-Hopf factorization of Markov Chains*, Ann. Appl. Probab., 4 (1994), pp. 390–413.
[29] G. W. Stewart, *Afternotes on Numerical Analysis*, SIAM, Philadelphia, 1996.
[30] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1358–1370

A NOTE ON BACKWARD ERROR ANALYSIS OF THE GENERALIZED SINGULAR VALUE DECOMPOSITION*

XIAO-SHAN CHEN† AND WEN LI†

Abstract. In terms of the generalized singular value decomposition, we define the generalized singular matrix sets and their backward errors. The explicit expressions of backward errors are derived, which extend a result of Sun [*SIAM J. Matrix Anal. Appl.*, 22 (2000), pp. 323–341]. The results are illustrated by a numerical example.

Key words. generalized singular matrix set, backward error, spectral norm, Frobenius norm

AMS subject classifications. 65F15, 65F99

DOI. 10.1137/070703351

1. Introduction.

The generalized singular value decomposition (GSVD) of two matrices having the same number of columns, first proposed by Van Loan [13], is a very useful tool in many matrix computation problems. Numerical methods and perturbation analysis of the GSVD have been developed (e.g., see [1, 2, 4, 8, 9, 10, 12]). In this paper, we discuss the backward errors for the GSVD and derive their explicit expressions.

Throughout this paper, we always use the following notations. Let $\mathcal{C}^{m\times n}$ be the set of $m \times n$ complex matrices, and let $\mathcal{C}^m = \mathcal{C}^{m\times 1}$ and $\mathcal{C} = \mathcal{C}^1$. The symbol I_p stands for the identity matrix of order p. By A^T, A^*, and $A^\dagger$ we denote the transpose, conjugate transpose, and Moore–Penrose inverse of a matrix A, respectively. $P_A = AA^\dagger$ is the orthogonal projection onto the column space of A and $P_A^\perp = I - P_A$. We use $\|\cdot\|_2, \|\cdot\|_F$, and $\|\cdot\|_\infty$ for the Euclidean vector norm and the spectral norm, Frobenius norm, and ∞-norm, respectively.

Let $A \in \mathcal{C}^{m\times n}$ and $B \in \mathcal{C}^{p\times n}$. The matrix pair $\{A, B\}$ is called an (m, p, n)-Grassmann matrix pair (GMP) if $\operatorname{rank}(A^T, B^T) = n$. In [6] Paige and Saunders obtained the following GSVD of the (m, p, n)-GMP.

THEOREM 1.1. $\{A, B\}$ (m, p, n) $U \in \mathcal{R}^{m\times m}, V \in \mathcal{R}^{p\times p}$ $X \in \mathcal{R}^{n\times n}$

$$U^*AX = \begin{pmatrix} D_\alpha & & \\ & 0_{(m-r-s)\times(n-r-s)} & \end{pmatrix}, \quad V^*BX = \begin{pmatrix} 0_{(p+r-n)\times r} & & \\ & & D_\beta \end{pmatrix}, \tag{1.1}$$

$0_{k\times l}$ $k \times l$

$$D_\alpha = \operatorname{diag}(\alpha_1, \ldots, \alpha_{r+s}), \quad D_\beta = \operatorname{diag}(\beta_{r+1}, \ldots, \beta_n) \tag{1.2}$$

$$1 = \alpha_1 = \cdots = \alpha_r > \alpha_{r+1} \geq \cdots \geq \alpha_{r+s} > \alpha_{r+s+1} = \cdots = \alpha_n = 0, \tag{1.3}$$

$$0 = \beta_1 = \cdots = \beta_r < \beta_{r+1} \leq \cdots \leq \beta_{r+s} < \beta_{r+s+1} = \cdots = \beta_n = 1, \tag{1.4}$$

*Received by the editors September 21, 2007; accepted for publication (in revised form) by B. T. Kågström May 19, 2008; published electronically October 16, 2008. This work was supported by the Natural Science Foundation of Guangdong Province (06025061) and by the National Natural Science Foundation of China (10671077).

http://www.siam.org/journals/simax/30-4/70335.html

†School of Mathematical Sciences, South China Normal University, Guangzhou, 510631, People's Republic of China (chenxs33@163.com, liwen@scnu.edu.cn).

$$\alpha_j^2 + \beta_j^2 = 1 \quad \forall j. \tag{1.5}$$

Let $\{A, B\}$ be an (m, p, n)-GMP with the GSVD given by (1.1)–(1.5). Then $\{(\alpha_j, \beta_j)\}_{j=1}^n$ are the generalized singular values (GSVs) of $\{A, B\}$, and every column x_j of the matrix X of (1.1) is a right generalized singular vector of $\{A, B\}$ associated with (α_j, β_j).

Let l be a natural number satisfying $\max\{n-p, 0\} < l < \min\{m, n\}$. From (1.1)–(1.5) we obtain formally

$$A(X_1, X_2) = (U_1, U_2)\begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}, \quad B(X_1, X_2) = (V_1, V_2)\begin{pmatrix} B_{11} & 0 \\ 0 & B_{22} \end{pmatrix} \tag{1.6}$$

and

$$A^*(U_1, U_2) = (Y_1, Y_2)\begin{pmatrix} A_{11}^* & 0 \\ 0 & A_{22}^* \end{pmatrix}, \quad B^*(V_1, V_2) = (Y_1, Y_2)\begin{pmatrix} B_{11}^* & 0 \\ 0 & B_{22}^* \end{pmatrix}, \tag{1.7}$$

where $A_{11} \in \mathcal{C}^{l\times l}$, $B_{11} \in \mathcal{C}^{(p+l-n)\times l}$, and the matrices $(U_1, U_2) \in \mathcal{C}^{m\times m}$, $(V_1, V_2) \in \mathcal{C}^{p\times p}$ with $U_1 \in \mathcal{C}^{m\times l}$, $V_1 \in \mathcal{C}^{p\times(p+l-n)}$ are unitary, and $X = (X_1, X_2) \in \mathcal{C}^{n\times n}$, $Y = X^{-*} = (Y_1, Y_2)$ with X_1, $Y_1 \in \mathcal{C}^{n\times l}$ are nonsingular.

From the relations (1.6) and (1.7) we have

$$\begin{aligned} &AX_1 = U_1A_{11}, \quad BX_1 = V_1B_{11}, \quad A^*U_1 = Y_1A_{11}^*, \quad B^*V_1 = Y_1B_{11}^*, \\ &Y_1^*X_1 = U_1^*U_1 = I_l, \quad V_1^*V_1 = I_{p+l-n}. \end{aligned} \tag{1.8}$$

The matrix set $\{X_1, Y_1, U_1, V_1\}$ of (1.8) is called a generalized singular matrix set of $\{A,\ B\}$ associated with the $(l, p+l-n, l)$-GMP $\{A_{11}, B_{11}\}$.

Let $\{\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1\}$ be an approximate generalized singular matrix set of $\{A,\ B\}$ associated with an approximate $(l, p+l-n, l)$-GMP $\{\widetilde{A}_{11}, \widetilde{B}_{11}\}$; that is, the matrices $\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1$ and $\widetilde{V}_1$ satisfy

$$\begin{aligned} &A\,\widetilde{X}_1 \approx \widetilde{U}_1\widetilde{A}_{11}, \quad B\,\widetilde{X}_1 \approx \widetilde{V}_1\widetilde{B}_{11}, \quad A^*\,\widetilde{U}_1 \approx \widetilde{Y}_1\widetilde{A}_{11}^*, \quad B^*\,\widetilde{V}_1 \approx \widetilde{Y}_1\widetilde{B}_{11}^*, \\ &\widetilde{Y}_1^*\widetilde{X}_1 = \widetilde{U}_1^*\widetilde{U}_1 = I_l, \quad \widetilde{V}_1^*\widetilde{V}_1 = I_{p+l-n}. \end{aligned}$$

We always assume that γ_A and γ_B are positive parameters.

Now we define the set $\mathcal{E}_1$ by

$$\mathcal{E}_1 = \left\{ \begin{pmatrix} E \\ F \end{pmatrix} : \begin{array}{ll} (A+E)\,\widetilde{X}_1 = \widetilde{U}_1\widetilde{A}_{11}, & (A+E)^*\,\widetilde{U}_1 = \widetilde{Y}_1\widetilde{A}_{11}^*, \\ (B+F)\,\widetilde{X}_1 = \widetilde{V}_1\widetilde{B}_{11}, & (B+F)^*\,\widetilde{V}_1 = \widetilde{Y}_1\widetilde{B}_{11}^* \end{array} \right\}, \tag{1.9}$$

and define the backward errors $\eta_2(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1)$ and $\eta_F(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1)$ of $\{A,\ B\}$ with respect to the approximate generalized singular matrix set $\{\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1\}$ associated with the approximate $(l, p+l-n, l)$-GMP $\{\widetilde{A}_{11}, \widetilde{B}_{11}\}$ by

$$\eta_2(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) = \min_{\left(\begin{smallmatrix} E \\ F \end{smallmatrix}\right) \in \mathcal{E}_1} \left\| \left(\frac{\|E\|_2}{\gamma_A}, \frac{\|F\|_2}{\gamma_B} \right) \right\|_\infty \tag{1.10}$$

and

$$\eta_F(\{\widetilde{A}_{11},\widetilde{B}_{11}\},\widetilde{X}_1,\widetilde{Y}_1,\widetilde{U}_1,\widetilde{V}_1)=\min_{\left(\begin{smallmatrix}E\\F\end{smallmatrix}\right)\in\mathcal{E}_1}\left\|\left(\frac{E}{\gamma_A},\ \frac{F}{\gamma_B}\right)\right\|_F, \tag{1.11}$$

respectively.

In some applications (e.g., investigating the generalized singular subspace group [4, 8, 10]) we only need an approximate matrix set $\{\widetilde{X}_1,\widetilde{Y}_1,\widetilde{U}_1,\widetilde{V}_1\}$, while $\{\widetilde{A}_{11},\widetilde{B}_{11}\}$ is not available. Hence we define the backward error $\eta_F(\widetilde{X}_1,\widetilde{Y}_1,\widetilde{U}_1,\widetilde{V}_1)$ of $\{A,\ B\}$ with respect to the approximate generalized singular matrix set $\{\widetilde{X}_1,\widetilde{Y}_1,\widetilde{U}_1,\widetilde{V}_1\}$ by

$$\eta_F(\widetilde{X}_1,\widetilde{Y}_1,\widetilde{U}_1,\widetilde{V}_1)=\min_{\left(\begin{smallmatrix}E\\F\end{smallmatrix}\right)\in\mathcal{E}_2}\left\|\left(\frac{E}{\gamma_A},\frac{F}{\gamma_B}\right)\right\|_F, \tag{1.12}$$

where

$$\mathcal{E}_2=\left\{\begin{pmatrix}E\\F\end{pmatrix}:\ \begin{array}{ll}(A+E)\ \widetilde{X}_1=\widetilde{U}_1\widetilde{A}_1, & (A+E)^*\ \widetilde{U}_1=\widetilde{Y}_1\widetilde{A}_1^*,\\ (B+F)\ \widetilde{X}_1=\widetilde{V}_1\widetilde{B}_1, & (B+F)^*\ \widetilde{V}_1=\widetilde{Y}_1\widetilde{B}_1^*,\\ \{\widetilde{A}_1,\widetilde{B}_1\}\text{ is an }(l,p+l-n,l)\text{-GMP}\end{array}\right\}. \tag{1.13}$$

It is well known that numerical algorithms for computing the GSVD do not produce the factors X_1 and Y_1 simultaneously (e.g., see [1, 5]). So, one needs to invert Y^* in order to get X_1. Hence if Y is very ill-conditioned, this inversion will introduce additional errors. Hence we also define the backward error $\eta_F(\{\widetilde{A}_{11},\widetilde{B}_{11}\},\widetilde{Y}_1,\widetilde{U}_1,\widetilde{V}_1)$ with respect to an approximate generalized singular matrix set $\{\widetilde{Y}_1,\widetilde{U}_1,\widetilde{V}_1\}$ associated with an approximate $(l,p+l-n,l)$-GMP $\{\widetilde{A}_{11},\widetilde{B}_{11}\}$ by

$$\eta_F(\{\widetilde{A}_{11},\widetilde{B}_{11}\},\widetilde{Y}_1,\widetilde{U}_1,\widetilde{V}_1)=\min_{\left(\begin{smallmatrix}E\\F\end{smallmatrix}\right)\in\mathcal{E}_3}\left\|\left(\frac{E}{\gamma_A},\frac{F}{\gamma_B}\right)\right\|_F, \tag{1.14}$$

where

$$\mathcal{E}_3=\left\{\begin{pmatrix}E\\F\end{pmatrix}:\ (A+E)^*\ \widetilde{U}_1=\widetilde{Y}_1\widetilde{A}_{11}^*,\quad (B+F)^*\ \widetilde{V}_1=\widetilde{Y}_1\widetilde{B}_{11}^*\right\}. \tag{1.15}$$

Similarly to (1.12) we also define the backward error $\eta_F(\widetilde{Y}_1,\widetilde{U}_1,\widetilde{V}_1)$ of $\{A,B\}$ with respect to the approximate generalized singular matrix set $\{\widetilde{Y}_1,\widetilde{U}_1,\widetilde{V}_1\}$ by

$$\eta_F(\widetilde{Y}_1,\widetilde{U}_1,\widetilde{V}_1)=\min_{\left(\begin{smallmatrix}E\\F\end{smallmatrix}\right)\in\mathcal{E}_4}\left\|\left(\frac{E}{\gamma_A},\frac{F}{\gamma_B}\right)\right\|_F, \tag{1.16}$$

where

$$\mathcal{E}_4=\left\{\begin{pmatrix}E\\F\end{pmatrix}:\ \begin{array}{ll}(A+E)^*\ \widetilde{U}_1=\widetilde{Y}_1\widetilde{A}_1^*, & (B+F)^*\ \widetilde{V}_1=\widetilde{Y}_1\widetilde{B}_1^*,\\ \{\widetilde{A}_1,\widetilde{B}_1\}\text{ is an }(l,p+l-n,l)\text{-GMP}\end{array}\right\}. \tag{1.17}$$

1.1. Taking $\gamma_A = \gamma_B = 1$, the above errors are called the absolute backward errors; and taking $\gamma_A = \|A\|_F$ and $\gamma_B = \|B\|_F$ in (1.11), (1.12), (1.14), and (1.16), and taking $\gamma_A = \|A\|_2$ and $\gamma_B = \|B\|_2$ in (1.10), the above errors are called the relative backward errors.

1.2. By definitions of $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$, and $\mathcal{E}_4$, it is easy to see that

$$\mathcal{E}_1 \subset \mathcal{E}_2, \quad \mathcal{E}_3 \subset \mathcal{E}_4, \quad \mathcal{E}_1 \subset \mathcal{E}_3, \quad \mathcal{E}_2 \subset \mathcal{E}_4.$$

Hence we have

$$\eta_F(\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) \le \eta_F(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1),$$

$$\eta_F(\widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) \le \eta_F(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1),$$

$$\eta_F(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) \le \eta_F(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1),$$

and

$$\eta_F(\widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) \le \eta_F(\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1).$$

From the definitions of the backward errors, we know that a small backward error means that the approximate solution of a problem is the exact one of a slightly perturbed problem. For example, a small $\eta_F(\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1)$ means that the approximate generalized singular matrix set $\{\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1\}$ is the exact generalized singular matrix set of a slightly perturbed $\{\widetilde{A}, \widetilde{B}\}$ of $\{A, B\}$. Consequently, a computable formula of the backward error may be useful for assessing the numerical quality of a computed GSVD and for testing the backward stability of algorithms for the computation of the GSVD.

The rest of this paper is organized as follows. In section 2, we shall derive explicit expressions of the backward errors of $\eta_2(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1)$, $\eta_F(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1)$, $\eta_F(\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1)$, $\eta_F(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{U}_1, \widetilde{V}_1)$, and $\eta_F(\widetilde{X}_1, \widetilde{U}_1, \widetilde{V}_1)$, respectively. We extend a result of Sun [12] to the generalized singular matrix set. In section 3, the results are illustrated by a numerical example.

2. Expressions of backward errors. In order to obtain the explicit expressions of backward errors for the GSVD, we need the following lemmas.

LEMMA 2.1 (see [12]). Let $A \in \mathcal{C}^{k\times m}, B \in \mathcal{C}^{n\times l}$, and $C \in \mathcal{C}^{k\times l}$. Define

$$\mathcal{E} = \{E \in \mathcal{C}^{m\times n} : \ AEB = C\}.$$

Then $\mathcal{E} \neq \emptyset$ (i.e., the matrix equation $AEB = C$ is consistent) for any A, B, and C if and only if

$$P_A C P_{B^*} = C,$$

where P_A and P_{B^*} denote the orthogonal projections onto the range of A and the range of B^*, respectively. In this case, the general solution of $AEB = C$ can be expressed as

$$E = A^\dagger C B^\dagger + Z - P_{A^*} Z P_B, \quad Z \in \mathcal{C}^{m\times n}.$$

LEMMA 2.2. Let $A \in \mathcal{C}^{m\times n}, U_1 \in \mathcal{C}^{m\times l}, A_1 \in \mathcal{C}^{l\times r}$, and $X_1, Y_1 \in \mathcal{C}^{n\times r}$ with $U_1^* U_1 = I_l, \operatorname{rank}(X_1) = r$. Then

$$\mathcal{G} = \left\{E \in \mathcal{C}^{m\times n} : \ (A+E)X_1 = U_1 A_1, \quad (A+E)^* U_1 = Y_1 A_1^*\right\}. \tag{2.1}$$

$X_1 = W_1 H$ X_1 $W_1^* W_1 = I_r$ H $U_2 \in \mathcal{C}^{m\times(m-l)}$, $W_2 \in \mathcal{C}^{n\times(n-r)}$ $U = (U_1, U_2)$ $W = (W_1, W_2)$ $\mathcal{G} \neq \emptyset$ $E \in \mathcal{G}$

$$E = U \begin{pmatrix} U_1^*(U_1A_1 - AX_1)H^{-1} & (A_1Y_1^* - U_1^*A)W_2 \\ -U_2^*AW_1 & U_2^*LW_2 \end{pmatrix} W^*, \quad L \in \mathcal{C}^{m\times n}. \tag{2.2}$$

. From (2.1), we know that $E \in \mathcal{G}$ if and only if E satisfies

$$EX_1 = U_1A_1 - AX_1, \quad E^*U_1 = Y_1A_1^* - A^*U_1. \tag{2.3}$$

Applying Lemma 2.1 to the first equation of (2.3), it is easy to see that the equation is solvable and any solution E can be expressed as

$$E = (U_1A_1 - AX_1)X_1^\dagger + Z(I - X_1X_1^\dagger), \quad Z \in \mathcal{C}^{m\times n}. \tag{2.4}$$

Since $X_1^\dagger = H^{-1}W_1^*$ and $X_1X_1^\dagger = W_1W_1^*$, (2.4) can be written as

$$E = (U_1A_1 - AX_1)H^{-1}W_1^* + ZW_2W_2^*, \tag{2.5}$$

which together with the second equation of (2.3) gives

$$W_1H^{-1}(U_1A_1 - AX_1)^*U_1 + W_2W_2^*Z^*U_1 = Y_1A_1^* - A^*U_1.$$

Multiplying the above equation by W_2^* on the left-hand side yields

$$U_1^*ZW_2 = (Y_1A_1^* - A^*U_1)^*W_2. \tag{2.6}$$

By Lemma 2.1, (2.6) is solvable, and any solution Z can be expressed as

$$Z = U_1(A_1Y_1^* - U_1^*A)W_2W_2^* + L - U_1U_1^*LW_2W_2^*, \quad L \in \mathcal{C}^{m\times n}. \tag{2.7}$$

Substituting (2.7) into (2.5) gives

$$E = (U_1A_1 - AX_1)H^{-1}W_1^* + U_1(A_1Y_1^* - U_1^*A)W_2W_2^* + U_2U_2^*LW_2W_2^*, \quad L \in \mathcal{C}^{m\times n},$$

from which (2.2) follows immediately. □

Sun [11] provided the following lemma, which can be found in [7].

LEMMA 2.3 (see [11]). . $F \in \mathcal{C}^{p\times m}, G \in \mathcal{C}^{n\times q}, and\ K \in \mathcal{C}^{p\times q}$ $X_* = F^\dagger KG^\dagger$.

$$\min_{X\in\mathcal{C}^{m\times n}} \|FXG - K\|_F = \|FX_*G - K\|_F.$$

LEMMA 2.4 (see [3]). .

$$f(X) = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & X \end{pmatrix}$$

$A_{11} \in \mathcal{C}^{k\times k}$ $A_{21}, A_{12}^T \in \mathcal{C}^{l\times k}$.

$$\min_{X\in\mathcal{C}^{l\times l}} \|f(X)\|_2 = \max\left\{ \left\| \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} \right\|_2, \ \|(A_{11},\ A_{12})\|_2 \right\}.$$

The following theorem gives explicit expressions of $\eta_2(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1)$ and $\eta_F(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1)$.

THEOREM 2.1. $\widetilde{U}_1 \in \mathcal{C}^{m\times l}, \widetilde{V}_1 \in \mathcal{C}^{p\times(p+l-n)}, \widetilde{X}_1, \widetilde{Y}_1 \in \mathcal{C}^{n\times l}, \widetilde{A}_{11} \in \mathcal{C}^{l\times l}$, $\widetilde{B}_{11} \in \mathcal{C}^{(p+l-n)\times l}$ $\widetilde{X}_1^* \widetilde{Y}_1 = \widetilde{U}_1^* \widetilde{U}_1 = I_l, \widetilde{V}_1^* \widetilde{V}_1 = I_{p+l-n}$. $\{\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1\}$ (m, p, n) $\{A, B\}$ $\{\widetilde{A}_{11}, \widetilde{B}_{11}\}$

$$\eta_2(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) \tag{2.8}$$
$$= \max\left\{ \frac{1}{\gamma_A} \max\left\{ \left\| R_1 \widetilde{X}_1^\dagger \right\|_2, \|R_3\|_2 \right\}, \frac{1}{\gamma_B} \max\left\{ \left\| R_2 \widetilde{X}_1^\dagger \right\|_2, \|R_4\|_2 \right\} \right\}$$

$$\eta_F(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) \tag{2.9}$$
$$= \sqrt{\begin{aligned} &\frac{1}{\gamma_A^2}\left(\|R_3\|_F^2 + \left\| R_1 \widetilde{X}_1^\dagger \right\|_F^2 - \left\| \widetilde{U}_1^* R_1 \widetilde{X}_1^\dagger \right\|_F^2 \right) \\ &\quad + \frac{1}{\gamma_B^2}\left(\|R_4\|_F^2 + \left\| R_2 \widetilde{X}_1^\dagger \right\|_F^2 - \left\| \widetilde{V}_1^* R_2 \widetilde{X}_1^\dagger \right\|_F^2 \right), \end{aligned}}$$

$$\begin{aligned} R_1 &= \widetilde{U}_1 \widetilde{A}_{11} - A \widetilde{X}_1, R_2 = \widetilde{V}_1 \widetilde{B}_{11} - B \widetilde{X}_1, \\ R_3 &= \widetilde{Y}_1 \widetilde{A}_{11}^* - A^* \widetilde{U}_1, \text{and } R_4 = \widetilde{Y}_1 \widetilde{B}_{11}^* - B^* \widetilde{V}_1. \end{aligned} \tag{2.10}$$

. From (1.9), we know that $\binom{E}{F} \in \mathcal{E}_1$ if and only if E and F satisfy

$$(A + E)\widetilde{X}_1 = \widetilde{U}_1 \widetilde{A}_{11}, \quad (A + E)^* \widetilde{U}_1 = \widetilde{Y}_1 \widetilde{A}_{11}^* \tag{2.11}$$

and

$$(B + F)\widetilde{X}_1 = \widetilde{V}_1 \widetilde{B}_{11}, \quad (B + F)^* \widetilde{V}_1 = Y_1 B_{11}^*. \tag{2.12}$$

Let $\widetilde{X}_1 = \widetilde{W}_1 \widetilde{H}$ be the polar decomposition of $\widetilde{X}_1$, where $\widetilde{W}_1^* \widetilde{W}_1 = I_r$ and $\widetilde{H}$ is a Hermitian positive definite matrix, and choose $\widetilde{U}_2 \in \mathcal{C}^{m\times(m-l)}, \widetilde{W}_2 \in \mathcal{C}^{n\times(n-l)}$, and $\widetilde{V}_2 \in \mathcal{C}^{p\times(n-l)}$ so that $\widetilde{U} = (\widetilde{U}_1, \widetilde{U}_2), \widetilde{W} = (\widetilde{W}_1, \widetilde{W}_2)$, and $\widetilde{V} = (\widetilde{V}_1, \widetilde{V}_2)$ are unitary. By (2.10) we have

$$\widetilde{U}_2^* R_1 \widetilde{H}^{-1} = -\widetilde{U}_2^* A \widetilde{W}_1 \quad \text{and} \quad \widetilde{V}_2^* R_2 \widetilde{H}^{-1} = -\widetilde{V}_2^* A \widetilde{W}_1.$$

Thus applying Lemma 2.2 to (2.11) and (2.12) gives

$$E = \widetilde{U} \begin{pmatrix} \widetilde{U}_1^* R_1 \widetilde{H}^{-1} & R_3^* \widetilde{W}_2 \\ \widetilde{U}_2^* R_1 \widetilde{H}^{-1} & \widetilde{U}_2^* L \widetilde{W}_2 \end{pmatrix} \widetilde{W}^* \equiv E(L), \quad L \in \mathcal{C}^{m\times n}, \tag{2.13}$$

and

$$F = \widetilde{V}\begin{pmatrix} \widetilde{V}_1^*\, R_2\, \widetilde{H}^{-1} & R_4^*\, \widetilde{W}_2 \\ \widetilde{V}_2^*\, R_2\, \widetilde{H}^{-1} & \widetilde{V}_2^*\, N\, \widetilde{W}_2 \end{pmatrix} \widetilde{W}^* \equiv F(N), \quad N \in \mathcal{C}^{p\times n}, \tag{2.14}$$

respectively, where R_1, R_2, R_3, and R_4 are given by (2.10). From Lemma 2.4 and (2.13) and (2.14), we obtain

$$\min_{L\in\mathcal{C}^{m\times n}} \frac{1}{\gamma_A}\|E(L)\|_2 = \frac{1}{\gamma_A}\max\left\{\left\|\begin{pmatrix} \widetilde{U}_1^*\, R_1\, \widetilde{H}^{-1} \\ \widetilde{U}_2^*\, R_1\, \widetilde{H}^{-1} \end{pmatrix}\right\|_2, \quad \left\|\left(\widetilde{U}_1^*\, R_1\, \widetilde{H}^{-1},\, R_3^*\, \widetilde{W}_2\right)\right\|_2\right\} \tag{2.15}$$

and

$$\min_{N\in\mathcal{C}^{p\times n}} \frac{1}{\gamma_B}\|F(N)\|_2 = \frac{1}{\gamma_B}\max\left\{\left\|\begin{pmatrix} \widetilde{V}_1^*\, R_2\, \widetilde{H}^{-1} \\ \widetilde{V}_2^*\, R_2\, \widetilde{H}^{-1} \end{pmatrix}\right\|_2, \quad \left\|\left(\widetilde{V}_1^*\, R_2\, \widetilde{H}^{-1},\, R_4^*\, \widetilde{W}_2\right)\right\|_2\right\}. \tag{2.16}$$

The equations $\widetilde{X}_1 = \widetilde{W}_1\widetilde{H}$ and $\widetilde{Y}_1^*\widetilde{X}_1 = I$ imply that $\widetilde{X}_1\widetilde{H}^{-1} = \widetilde{W}_1$ and $\widetilde{H}^{-1} = \widetilde{Y}_1^*\widetilde{W}_1$, which together with (2.10) gives

$$\widetilde{U}_1^*\, R_1\, \widetilde{H}^{-1} = R_3^*\, \widetilde{W}_1, \quad \widetilde{V}_1^*\, R_2\, \widetilde{H}^{-1} = R_4^*\, \widetilde{W}_1. \tag{2.17}$$

Clearly,

$$\left\|R_1\, \widetilde{H}^{-1}\right\|_2 = \left\|R_1\, \widetilde{X}_1^{\dagger}\right\|_2, \quad \left\|R_2\, \widetilde{H}^{-1}\right\|_2 = \left\|R_2\, \widetilde{X}_1^{\dagger}\right\|_2. \tag{2.18}$$

Combining (1.10) with (2.15)–(2.18) yields (2.8).

Now we prove (2.9). By (1.11), (2.13), (2.14), and (2.17) we have

$$\begin{aligned}
&\eta_F^2(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) \\
&\quad = \min_{L\in\mathcal{C}^{m\times n}} \frac{1}{\gamma_A^2}\,\|E(L)\|_F^2 + \min_{N\in\mathcal{C}^{p\times n}} \frac{1}{\gamma_B^2}\,\|F(N)\|_F^2 \\
&\quad = \frac{1}{\gamma_A^2}\left(\left\|\widetilde{U}_1^*\, R_1\, \widetilde{H}^{-1}\right\|_F^2 + \left\|\widetilde{U}_2^*\, R_1\, \widetilde{H}^{-1}\right\|_F^2 + \left\|R_3^*\, \widetilde{W}_2\right\|_F^2\right) \\
&\qquad + \frac{1}{\gamma_B^2}\left(\left\|\widetilde{V}_1^*\, R_2\, \widetilde{H}^{-1}\right\|_F^2 + \left\|\widetilde{V}_2^*\, R_2\, \widetilde{H}^{-1}\right\|_F^2 + \left\|R_4^*\, \widetilde{W}_2\right\|_F^2\right) \\
&\quad = \frac{1}{\gamma_A^2}\left(\left\|R_1\, \widetilde{H}^{-1}\right\|_F^2 + \|R_3\|_F^2 - \left\|\widetilde{U}_1^*\, R_1\, \widetilde{H}^{-1}\right\|_F^2\right) \\
&\qquad + \frac{1}{\gamma_B^2}\left(\left\|R_2\, \widetilde{H}^{-1}\right\|_F^2 + \|R_4\|_F^2 - \left\|\widetilde{V}_1^*\, R_2\, \widetilde{H}^{-1}\right\|_F^2\right),
\end{aligned}$$

from which (2.9) follows immediately. □

2.1. It is noted that the result of Sun [12] can be derived from (2.8). Let $(\widetilde{\alpha},\ \widetilde{\beta}) \neq (0,\ 0)$ with $\widetilde{\alpha}, \widetilde{\beta} \geq 0$ be an approximate GSV of an (m,p,n)-GMP$\{A,\ B\}$, and let $\{\widetilde{x}, \widetilde{y}, \widetilde{u}, \widetilde{v}\}$ be an associated approximate generalized singular vector set, that is, the vectors $\widetilde{x}, \widetilde{y} \in \mathcal{C}^n, \widetilde{u} \in \mathcal{C}^m$, and $\widetilde{v} \in \mathcal{C}^p$ satisfy

$$A\ \widetilde{x} \approx \widetilde{\alpha}\widetilde{u}, \quad B\ \widetilde{x} \approx \widetilde{\beta}\widetilde{v}, \quad A^*\ \widetilde{u} \approx \widetilde{\alpha}\widetilde{y}, \quad B^*\ \widetilde{v} \approx \widetilde{\beta}\widetilde{y},$$

$$\widetilde{y}^*\widetilde{x} = 1, \quad \|\ \widetilde{u}\ \|_2 = \|\ \widetilde{v}\ \|_2 = 1.$$

Let

$$r_1 = \widetilde{\alpha}\widetilde{u} - A\ \widetilde{x}, \qquad r_2 = \widetilde{\beta}\widetilde{v} - B\ \widetilde{x},$$

$$r_3 = \widetilde{\alpha}\widetilde{y} - A^*\ \widetilde{u}, \qquad r_4 = \widetilde{\beta}\widetilde{y} - B^*\ \widetilde{v}.$$

Then by (2.8) and (2.9) we can obtain

$$\eta_2((\widetilde{\alpha},\widetilde{\beta}),\widetilde{x},\widetilde{y},\widetilde{u},\widetilde{v})$$

$$= \max\left\{\frac{1}{\gamma_A}\max\left\{\frac{\|r_1\|_2}{\|\ \widetilde{x}\ \|_2},\ \|r_3\|_2\right\}, \frac{1}{\gamma_B}\max\left\{\frac{\|r_2\|_2}{\|\ \widetilde{x}\ \|_2},\ \|r_4\|_2\right\}\right\},$$

which was proved by Sun [12] and

$$\eta_F((\widetilde{\alpha},\widetilde{\beta}),\widetilde{x},\widetilde{y},\widetilde{u},\widetilde{v})$$

$$= \sqrt{\frac{1}{\gamma_A^2}\left(\|r_3\|_2^2 + \frac{\|r_1\|_2^2}{\|\ \widetilde{x}\ \|_2^2} - \frac{|\ \widetilde{u}^*\ r_1|^2}{\|\ \widetilde{x}\ \|_2^2}\right) + \frac{1}{\gamma_B^2}\left(\|r_4\|_2^2 + \frac{\|r_2\|_2^2}{\|\ \widetilde{x}\ \|_2^2} - \frac{|\ \widetilde{v}^*\ r_2|^2}{\|\ \widetilde{x}\ \|_2^2}\right)},$$

respectively.

Next we give the expression of $\eta_F(\ \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1\ )$.

THEOREM 2.2. $\widetilde{U}_1 \in \mathcal{C}^{m\times l}, \widetilde{V}_1 \in \mathcal{C}^{p\times(p+l-n)}$ $\widetilde{X}_1, \widetilde{Y}_1 \in \mathcal{C}^{n\times l}$ $\widetilde{U}_1^*\widetilde{U}_1 = \widetilde{X}_1^*\widetilde{Y}_1 = I_l$ $\widetilde{V}_1^*\widetilde{V}_1 = I_{p+l-n}$ $\{\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1\}$ (m,p,n) $GMP\{A,\ B\}$ $\{\widetilde{U}_1^*\ A\ \widetilde{Y}_1^{\dagger *},\ \widetilde{V}_1^*\ B\ \widetilde{Y}_1^{\dagger *}\}$ $(l, p+l-n, l)$

$$\eta_F(\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) \tag{2.19}$$

$$= \sqrt{\frac{1}{\gamma_A^2}\left(\left\|P_{\widetilde{U}_1}^{\perp} A P_{\widetilde{X}_1}\right\|_F^2 + \left\|P_{\widetilde{U}_1} A P_{\widetilde{Y}_1}^{\perp}\right\|_F^2\right) + \frac{1}{\gamma_B^2}\left(\left\|P_{\widetilde{V}_1}^{\perp} B P_{\widetilde{X}_1}\right\|_F^2 + \left\|P_{\widetilde{V}_1} B P_{\widetilde{Y}_1}^{\perp}\right\|_F^2\right)}.$$

. From (1.13), we know that $\binom{E}{F} \in \mathcal{E}_2$ if and only if there exists an $(l, p+l-n, l)$-GMP $\{\widetilde{A}_1,\ \widetilde{B}_1\}$ such that

$$(A+E)\ \widetilde{X}_1 = \widetilde{U}_1\widetilde{A}_1, \quad (A+E)^*\ \widetilde{U}_1 = \widetilde{Y}_1\widetilde{A}_1^* \tag{2.20}$$

and

$$(B+F)\,\widetilde{X}_1=\widetilde{V}_1\widetilde{B}_1,\quad (B+F)^*\,\widetilde{V}_1=\widetilde{Y}_1\widetilde{B}_1^*\,. \tag{2.21}$$

Applying Lemma 2.2 to (2.20) and (2.21) gives

$$\begin{aligned} E &= \widetilde{U}\begin{pmatrix} \widetilde{U}_1^*\,(\widetilde{U}_1\widetilde{A}_1-A\,\widetilde{X}_1)\,\widetilde{H}^{-1} & (\widetilde{A}_1\widetilde{Y}_1^*-\widetilde{U}_1^*\,A)\,\widetilde{W}_2 \\ -\,\widetilde{U}_2^*\,A\,\widetilde{W}_1 & \widetilde{U}_2^*\,L\,\widetilde{W}_2 \end{pmatrix}\widetilde{W}^* \\ &\equiv E\left(\widetilde{A}_1,\ L\right),\quad L\in\mathcal{C}^{m\times n}, \end{aligned} \tag{2.22}$$

and

$$\begin{aligned} F &= \widetilde{V}\begin{pmatrix} \widetilde{V}_1^*\,(\widetilde{V}_1\widetilde{B}_1-B\,\widetilde{X}_1)\,\widetilde{H}^{-1} & (\widetilde{B}_1\widetilde{Y}_1^*-\widetilde{V}_1^*\,B)\,\widetilde{W}_2 \\ -\,\widetilde{V}_2^*\,B\,\widetilde{W}_1 & \widetilde{V}_2^*\,N\,\widetilde{W}_2 \end{pmatrix}\widetilde{W}^* \\ &\equiv F\left(\widetilde{B}_1,N\right),\quad N\in\mathcal{C}^{p\times n}, \end{aligned} \tag{2.23}$$

respectively, where $\widetilde{U}=(\widetilde{U}_1,\widetilde{U}_2),\widetilde{V}=(\widetilde{V}_1,\widetilde{V}_2),\widetilde{W}=(\widetilde{W}_1,\widetilde{W}_2)$, and $\widetilde{H}$ are given by the proof in Theorem 2.1. From (2.22) and (2.23), we have

$$\begin{aligned} \min_{\widetilde{A}_1\in\mathcal{C}^{l\times l},\ L\in\mathcal{C}^{m\times n}}\left\|E\left(\widetilde{A}_1,\ L\right)\right\|_F^2 = \min_{\widetilde{A}_1\in\mathcal{C}^{l\times l}}\Bigg(&\left\|\widetilde{U}_1^*\left(\widetilde{U}_1\widetilde{A}_1-A\,\widetilde{X}_1\right)\widetilde{H}^{-1}\right\|_F^2 \\ &+\left\|\left(\widetilde{A}_1\widetilde{Y}_1^*-\widetilde{U}_1^*\,A\right)\widetilde{W}_2\right\|_F^2\Bigg)+\left\|\widetilde{U}_2^*\,A\,\widetilde{W}_1\right\|_F^2 \end{aligned} \tag{2.24}$$

and

$$\begin{aligned} \min_{\widetilde{B}_1\in\mathcal{C}^{(p+l-n)\times l},\ N\in\mathcal{C}^{p\times n}}\left\|F\left(\widetilde{B}_1,\ L\right)\right\|_F^2 = \min_{\widetilde{B}_1\in\mathcal{C}^{(p+l-n)\times l}}\Bigg(&\left\|\widetilde{V}_1^*\left(\widetilde{V}_1\widetilde{B}_1-B\,\widetilde{X}_1\right)\widetilde{H}^{-1}\right\|_F^2 \\ &+\left\|\left(\widetilde{B}_1\widetilde{Y}_1^*-\widetilde{V}_1^*\,B\right)\widetilde{W}_2\right\|_F^2\Bigg)+\left\|\widetilde{V}_2^*\,B\,\widetilde{W}_1\right\|_F^2. \end{aligned} \tag{2.25}$$

It is easy to see that

$$\left\|\widetilde{U}_2^*\,A\,\widetilde{W}_1\right\|_F=\left\|P_{\widetilde{U}_1}^{\perp}\,AP_{\widetilde{X}_1}\right\|_F,\quad \left\|\widetilde{V}_2^*\,B\,\widetilde{W}_1\right\|_F=\left\|P_{\widetilde{V}_1}^{\perp}\,BP_{\widetilde{X}_1}\right\|_F. \tag{2.26}$$

Similarly to (2.17), we have

$$\widetilde{U}_1^*\left(\widetilde{U}_1\widetilde{A}_1-A\,\widetilde{X}_1\right)\widetilde{H}^{-1}=\left(\widetilde{A}_1\widetilde{Y}_1^*-\widetilde{U}_1^*\,A\right)\widetilde{W}_1, \tag{2.27}$$

$$\widetilde{V}_1^*\left(\widetilde{V}_1\widetilde{B}_1-B\,\widetilde{X}_1\right)\widetilde{H}^{-1}=\left(\widetilde{B}_1\widetilde{Y}_1^*-\widetilde{V}_1^*\,B\right)\widetilde{W}_1\,. \tag{2.28}$$

Hence from (2.27) and (2.28) we obtain

$$\left\|\widetilde{U}_1^*\,(\widetilde{U}_1\widetilde{A}_1 - A\,\widetilde{X}_1)\,\widetilde{H}^{-1}\right\|_F^2 + \left\|(\widetilde{A}_1\widetilde{Y}_1^* - \widetilde{U}_1^*\,A)\,\widetilde{W}_2\right\|_F^2 = \left\|\widetilde{A}_1\widetilde{Y}_1^* - \widetilde{U}_1^*\,A\right\|_F^2, \tag{2.29}$$

$$\left\|\widetilde{V}_1^*\,(\widetilde{V}_1\widetilde{B}_1 - B\,\widetilde{X}_1)\,\widetilde{H}^{-1}\right\|_F^2 + \left\|(\widetilde{B}_1\widetilde{Y}_1^* - \widetilde{V}_1^*\,B)\,\widetilde{W}_2\right\|_F^2 = \left\|\widetilde{B}_1\widetilde{Y}_1^* - \widetilde{V}_1^*\,B\right\|_F^2. \tag{2.30}$$

Applying Lemma 2.3 to (2.29) and (2.30) gives

$$\min_{\widetilde{A}_1\in\mathcal{C}^{l\times l}}\left\|\widetilde{A}_1\widetilde{Y}_1^* - \widetilde{U}_1^*\,A\right\|_F = \left\|\widetilde{U}_1^*\,A\,\widetilde{Y}_1^{*\dagger}\,\widetilde{Y}_1^* - \widetilde{U}_1^*\,A\right\|_F = \left\|P_{\widetilde{U}_1}AP_{\widetilde{Y}_1}^{\perp}\right\|_F \tag{2.31}$$

and

$$\min_{\widetilde{B}_1\in\mathcal{C}^{(p+l-n)\times l}}\left\|\widetilde{B}_1\widetilde{Y}_1^* - \widetilde{V}_1^*\,B\right\|_F = \left\|\widetilde{V}_1^*\,B\,\widetilde{Y}_1^{*\dagger}\,\widetilde{Y}_1^* - \widetilde{V}_1^*\,B\right\|_F = \left\|P_{\widetilde{V}_1}BP_{\widetilde{Y}_1}^{\perp}\right\|_F, \tag{2.32}$$

respectively. Notice that $\{\widetilde{U}_1^*\,A\,\widetilde{Y}_1^{\dagger *}, \widetilde{V}_1^*\,B\,\widetilde{Y}_1^{\dagger *}\}$ is an $(l, p+l-n, l)$-GMP. Combining (1.12) with (2.24)–(2.26) and (2.29)–(2.32) shows the formula (2.19). The proof is complete. □

The following two results give computable formulae of the backward errors $\eta_F(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1)$ and $\eta_F(\widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1)$.

THEOREM 2.3. $\widetilde{U}_1\in\mathcal{C}^{m\times l}, \widetilde{V}_1\in\mathcal{C}^{p\times(p+l-n)}, \widetilde{Y}_1\in\mathcal{C}^{n\times l}$, $\widetilde{A}_{11}\in\mathcal{C}^{l\times l}$, $\widetilde{B}_{11}\in\mathcal{C}^{(p+l-n)\times l}$, $\widetilde{U}_1^*\widetilde{U}_1 = I_l, \widetilde{V}_1^*\widetilde{V}_1 = I_{p+l-n}$. $\{\widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1\}$ (m, p, n) $\{A, B\}$ $\{\widetilde{A}_{11}, \widetilde{B}_{11}\}$

$$\eta_F(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) = \sqrt{\frac{1}{\gamma_A^2}\,\|R_3\|_F^2 + \frac{1}{\gamma_B^2}\,\|R_4\|_F^2}, \tag{2.33}$$

R_3 R_4 (2.10).

. From (1.15) and (2.10), we see that $\binom{E}{F}\in\mathcal{E}_3$ if and only if E and F satisfy

$$E^*\,\widetilde{U}_1 = R_3, \quad F^*\,\widetilde{V}_1 = R_4. \tag{2.34}$$

Applying Lemma 2.1 to (2.34) gives

$$E^* = R_3\,\widetilde{U}_1^* + Z_1(I_m - \widetilde{U}_1\widetilde{U}_1^*), \quad F^* = R_4\,\widetilde{V}_1^* + Z_2(I_p - \widetilde{V}_1\widetilde{V}_1^*),$$

where $Z_1\in\mathcal{C}^{m\times n}$ and $Z_2\in\mathcal{C}^{p\times n}$. From the above equations and (1.14), it is easy to see that

$$\begin{aligned}
&\eta_F^2(\{\widetilde{A}_{11}, \widetilde{B}_{11}\}, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1)\\
&= \frac{1}{\gamma_A^2}\min_{Z_1\in\mathcal{C}^{m\times n}}\left\|R_3\,\widetilde{U}_1^* + Z_1(I_m - \widetilde{U}_1\widetilde{U}_1^*)\right\|_F^2 + \frac{1}{\gamma_B^2}\min_{Z_2\in\mathcal{C}^{p\times n}}\left\|R_4\,\widetilde{V}_1^* + Z_2(I_n - \widetilde{V}_1\widetilde{V}_1^*)\right\|_F^2\\
&= \frac{1}{\gamma_A^2}\|R_3\|_F^2 + \frac{1}{\gamma_B^2}\|R_4\|_F^2,
\end{aligned}$$

which implies (2.33). The proof is complete. □

THEOREM 2.4. $\widetilde{U}_1 \in \mathcal{C}^{m\times l}, \widetilde{V}_1 \in \mathcal{C}^{p\times(p+l-n)}$ $\widetilde{Y}_1 \in \mathcal{C}^{n\times l}$ $\widetilde{U}_1^*\widetilde{U}_1 = I_l$, $\widetilde{V}_1^*\widetilde{V}_1 = I_{p+l-n}$ $\{\widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1\}$ (m,p,n) $\{A,\ B\}$ $\{\widetilde{U}_1^*\ A\ \widetilde{Y}_1^{\dagger *},\ \widetilde{V}_1^*\ B\ \widetilde{Y}_1^{\dagger *}\}$ $(l, p+l-n, l)$

$$\eta_F(\widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) = \sqrt{\frac{1}{\gamma_A^2}\left\|P_{\widetilde{U}_1} A P^{\perp}_{\widetilde{Y}_1}\right\|_F^2 + \frac{1}{\gamma_B^2}\left\|P_{\widetilde{V}_1} B P^{\perp}_{\widetilde{Y}_1}\right\|_F^2}. \tag{2.35}$$

. By definition (1.17), $\binom{E}{F} \in \mathcal{E}_4$ if and only if there is an $(l, p+l-n, l)$-GMP $\{\widetilde{A}_1, \widetilde{B}_1\}$ such that

$$E^*\ \widetilde{U}_1 = \widetilde{Y}_1\widetilde{A}_1^* - A^*\ \widetilde{U}_1, \quad F^*\ \widetilde{V}_1 = \widetilde{Y}_1\widetilde{B}_1^* - B^*\ \widetilde{V}_1\ .$$

By Lemma 2.1, we know that the above two equations are solvable and E and F are expressed by

$$E = \widetilde{U}_1\ (\widetilde{A}_1\widetilde{Y}_1^* - \widetilde{U}_1^*\ A) + (I_m - \widetilde{U}_1\widetilde{U}_1^*)Z_1,$$
$$F = \widetilde{V}_1\ (\widetilde{B}_1\widetilde{Y}_1^* - \widetilde{V}_1^*\ B) + (I_p - \widetilde{V}_1\widetilde{V}_1^*)Z_2,$$

where $Z_1 \in \mathcal{C}^{m\times n}, Z_2 \in \mathcal{C}^{p\times n}$. From (1.16) and the above two equalities we have

$$\begin{aligned}
\eta_F^2(\widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) &= \frac{1}{\gamma_A^2} \min_{\widetilde{A}_1\in\mathcal{C}^{l\times l}, Z_1\in\mathcal{C}^{m\times n}} \left\|\widetilde{U}_1\ (\widetilde{A}_1\widetilde{Y}_1^* - \widetilde{U}_1^*\ A) + (I_m - \widetilde{U}_1\widetilde{U}_1^*)Z_1\right\|_F^2 \\
&\quad + \frac{1}{\gamma_B^2} \min_{\widetilde{B}_1\in\mathcal{C}^{(p+l-n)\times l}, Z_2\in\mathcal{C}^{p\times n}} \left\|\widetilde{V}_1\ (\widetilde{B}_1\widetilde{Y}_1^* - \widetilde{V}_1^*\ B) + (I_p - \widetilde{V}_1\widetilde{V}_1^*)Z_2\right\|_F^2 \\
&= \frac{1}{\gamma_A^2} \min_{\widetilde{A}_1\in\mathcal{C}^{m\times n}} \left\|\widetilde{A}_1\widetilde{Y}_1^* - \widetilde{U}_1^*\ A\right\|_F^2 + \frac{1}{\gamma_B^2} \min_{\widetilde{B}_1\in\mathcal{C}^{(p+l-n)\times l}} \left\|\widetilde{B}_1\widetilde{Y}_1^* - \widetilde{V}_1^*\ B\right\|_F^2.
\end{aligned} \tag{2.36}$$

From Lemma 2.3 and (2.36) we have

$$\eta_F^2(\widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) = \frac{1}{\gamma_A^2}\left\|\widetilde{U}_1^*\ A(I - \widetilde{Y}_1\widetilde{Y}_1^{\dagger})\right\|_F^2 + \frac{1}{\gamma_B^2}\left\|\widetilde{V}_1^*\ B(I - \widetilde{Y}_1\widetilde{Y}_1^{\dagger})\right\|_F^2. \tag{2.37}$$

Notice that $\{\widetilde{U}_1^*\ A\ \widetilde{Y}_1^{\dagger *},\ \widetilde{V}_1^*\ B\ \widetilde{Y}_1^{\dagger *}\}$ is an $(l, p+l-n, l)$-GMP. Then (2.35) follows from (2.37). The proof is complete. □

3. A numerical example. In this section we use a simple example to illustrate the results of the previous section. All computations were performed by using MATLAB 6.5. The relative machine precision is 2.22×10^{-16}.

3.1. Consider the (3,3,3)-GMP $\{A, B\}$ with

$$A = \begin{pmatrix} \frac{7}{2} & 3 & -3 \\ 2\sqrt{2} & 4\sqrt{2}(1+10^{-14}) & -4\sqrt{2} \\ -\frac{1}{2} & 3 & -3 \end{pmatrix}, \quad B = \begin{pmatrix} -\frac{1}{2} & 1 & -1 \\ \frac{\sqrt{2}}{2} & \sqrt{2}(1+10^{-14}) & -\sqrt{2} \\ \frac{3}{2} & 1 & -1 \end{pmatrix}.$$

By using function **[U,V,Y,C,S] = gsvd(A,B)** in MATLAB 6.5, we get the computed GSVD $A = UCY^T$ and $B = VSY^T$, where

$$U = \begin{pmatrix} 0.72882766003858 & -0.68439687422507 & -0.02027709332328 \\ -0.01691197215049 & 0.01161155660932 & -0.99978955633228 \\ -0.68448829585368 & -0.72901720851024 & 0.00311167724380 \end{pmatrix},$$

$$V = \begin{pmatrix} -0.69179597797044 & -0.72206709187087 & 0.00611879898952 \\ -0.00818050022115 & -0.00063616783742 & -0.99996633678670 \\ 0.72204667735546 & -0.69182274473136 & -0.00546677165349 \end{pmatrix},$$

$$Y = \begin{pmatrix} 3.18138671232003 & -2.10966551430603 & -2.98799096458163 \\ 0.04176115800505 & -4.40778386049074 & -5.87960010926594 \\ -0.04176115800505 & 4.40778386049074 & 5.87960010926588 \end{pmatrix},$$

$$C = \operatorname{diag}(0.89436052092627, 0.94708758346252, 0.97067146447266),$$

and

$$S = \operatorname{diag}(0.44734691080692, 0.32097524710033, 0.24040987512682).$$

By using the function **inv** in MATLAB 6.5, we obtain

$$X = Y^{-T} = \begin{pmatrix} 0.31622982070089 & 0.00507812500000 & -0.00156084661164 \\ -9.288469999189297e+11 & 2.251799813685225e+13 & -1.688775709899456e+13 \\ -9.288469999187793e+11 & 2.251799813685248e+13 & -1.688775709899456e+13 \end{pmatrix}.$$

Let

$$\widetilde{A}_{11} = C(1:2,1:2), \quad \widetilde{B}_{11} = S(1:2,1:2), \quad \widetilde{\alpha}_3 = C(3,3), \quad \widetilde{\beta}_3 = S(3,3),$$

$$\widetilde{U}_1 = U(1:3,1:2), \quad \widetilde{u}_3 = U(1:3,3), \quad \widetilde{V}_1 = V(1:3,1:2), \quad \widetilde{v}_3 = V(1:3,3),$$

$$\widetilde{Y}_1 = Y(1:3,1:2), \quad \widetilde{y}_3 = Y(1:3,3), \quad \widetilde{X}_1 = X(1:3,1:2), \quad \widetilde{x}_3 = X(1:3,3),$$

where $K(i:j,k:l)$ is the submatrix of K with entries having row and column indices in the ranges i through j and k through l, respectively. Taking $\gamma_A = \|A\|_F$ and $\gamma_B = \|B\|_F$ we get the relative backward errors

$$\eta_F((\widetilde{A}_{11}, \widetilde{B}_{11}), \widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) = 0.00185341925320, \tag{3.1}$$

$$\eta_F((\widetilde{\alpha}_3, \widetilde{\beta}_3), \widetilde{x}_3, \widetilde{y}_3, \widetilde{u}_3, \widetilde{v}_3) = 8.724097354653458e-16, \tag{3.2}$$

$$\eta_F(\widetilde{X}_1, \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) = 0.00203991762337, \tag{3.3}$$

$$\eta_F(\widetilde{x}_3, \widetilde{y}_3, \widetilde{u}_3, \widetilde{v}_3) = 4.300498624031868e-16, \tag{3.4}$$

$$\eta_F((\widetilde{A}_{11}, \widetilde{B}_{11}), \widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) = 1.322954207342138e-15, \tag{3.5}$$

$$\eta_F((\widetilde{\alpha}_3, \widetilde{\beta}_3), \widetilde{y}_3, \widetilde{u}_3, \widetilde{v}_3) = 8.710521422021934e-16, \tag{3.6}$$

$$\eta_F(\widetilde{Y}_1, \widetilde{U}_1, \widetilde{V}_1) = 2.516175515362349e-16, \tag{3.7}$$

and

$$\eta_F(\widetilde{y}_3, \widetilde{u}_3, \widetilde{v}_3) = 4.255419420330271e-16. \tag{3.8}$$

The results (3.5)–(3.8) show that the computation of the GSVD by using function **gsvd** in MATLAB 6.5 has proceeded stably. But since Y is very ill-conditioned, in fact, $\mathrm{cond}_2(Y) = \|Y\|_2\|Y^{-1}\|_2 = 4.360846260436090e+014$, inverting Y to get X will introduce additional errors. This means that the backward errors of (3.1)–(3.4) may not be very small.

Acknowledgment. The authors would like to thank the anonymous referees for their valuable comments.

REFERENCES

[1] Z. Bai and J. W. Demmel, *Computing the generalized singular value decomposition*, SIAM J. Sci. Comput., 14 (1993), pp. 1464–1486.

[2] Z. Bai and H. Zha, *A new preprocessing algorithm for the computation of the generalized singular value decomposition*, SIAM J. Sci. Comput., 14 (1993), pp. 1007–1012.

[3] C. Davis, W. M. Kahan, and H. F. Weinberger, *Norm-preserving dilations and their applications to optimal error bounds*, SIAM J. Numer. Anal., 19 (1982), pp. 445–469.

[4] R. C. Li, *Bounds on perturbations of generalized singular values and of associated subspaces*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 195–234.

[5] C. C. Paige, *Computing the generalized singular value decomposition*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1126–1146.

[6] C. C. Paige and M. A. Saunders, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.

[7] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

[8] J. G. Sun, *Perturbation analysis for the generalized singular value problem*, SIAM J. Numer. Anal., 20 (1983), pp. 611–625.

[9] J. G. Sun, *Stability and Accuracy: Perturbation Analysis of Algebraic Eigenproblems*, Report UMINF 98.7, Department of Computing Science, Umea University, Umea, Sweden, 1998.

[10] J. G. Sun, *Perturbation analysis of generalized singular subspaces*, Numer. Math., 79 (1998), pp. 615–641.

[11] J. G. Sun, *Structured backward errors for KKT systems*, Linear Algebra Appl., 288 (1999), pp. 75–88.

[12] J.-G. Sun, *Condition number and backward error for the generalized singular value decomposition*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 323–341.

[13] C. F. Van Loan, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal., 13 (1976), pp. 76–83.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1371–1391

A GRAPH BASED APPROACH TO THE CONVERGENCE OF ONE LEVEL SCHWARZ ITERATIONS FOR SINGULAR M-MATRICES AND MARKOV CHAINS*

STEFAN BOROVAC†

Abstract. We study the convergence of additive and multiplicative Schwarz iterations applied to singular M-matrices and Markov chains. We do our investigations in order to solve consistent linear systems or to calculate a probability distribution vector of a Markov chain. It turns out that for a certain set of equations we are able to prove convergence for both methods with a reliable degree of freedom concerning the overlap. These new convergence theorems are based on a graph theoretical approach and represent the main results of this paper. Other applications of the introduced theory are also discussed.

Key words. linear systems, M-matrices, Markov chains, singular matrices, iterative methods, block methods, additive Schwarz, multiplicative Schwarz, domain decomposition methods, overlap, graph theory

AMS subject classifications. 05C50, 60J22, 65F10, 65F15, 65F35, 65M55

DOI. 10.1137/070687888

1. Introduction.

Schwarz methods are mainly used for the numerical solution of partial differential equations and can be classified as domain decomposition methods [22, 27, 30]. Another common application of these methods is to use them as a preconditioner for Krylov subspace methods. But recently, these methods have also been proposed to solve symmetric [20] and nonsymmetric [8, 16] Markov chains iteratively. To this purpose, an algebraic formulation was developed which is partly recapitulated in section 4; see also [3, 8, 13, 16, 19, 20].

In this paper we analyze Schwarz methods for the iterative solution of consistent linear systems of the form

$$Ax = b. \tag{1.1}$$

Here, either A is a square singular M-matrix or, as a special case, $A = I - B$, I the identity matrix and B a square column stochastic matrix [28]. We assume that (1.1) is consistent; i.e., b lies in the range of A.

We do not tackle the system (1.1) in the most general setting but introduce some restrictions concerning the null space of A. We assume that

$$\mathcal{N}(A) = \{x \in \mathbb{R}^n : Ax = 0\} = \operatorname{span}\{z\} \text{ for some positive vector } z. \tag{1.2}$$

This assumption allows us to construct convergent additive and multiplicative Schwarz iterations to solve (1.1). Whereas the convergence is not too hard to prove, the consistency of the convergence process takes considerable effort, and a lot of the theory developed here is to obtain consistency.

The theory presented can be applied to reducible nonsymmetric matrices A which satisfy (1.2). In [20] only symmetric semidefinite systems are considered. In [8] an

*Received by the editors April 11, 2007; accepted for publication (in revised form) by D. B. Szyld June 19, 2008; published electronically October 22, 2008.
http://www.siam.org/journals/simax/30-4/68788.html
†Fachbereich Mathematik und Naturwissenschaften, Universität Wuppertal, D-42097 Wuppertal, Germany (borovac@math.uni-wuppertal.de).

alternative to our ansatz is studied, but there only certain cases of overlap are considered. The approach in [16] appears to be more general than ours but relies on an assumption of consistency which is actually not known to hold.

Our theory is based on properties of the nonzero pattern of A which must necessarily exist if A satisfies (1.2) (cf. section 3). We use this in a graph based approach to construct convergent Schwarz iterations which are also consistent in the sense that we converge to a reliable solution. For one level multiplicative Schwarz we do this in two steps. First we present the basic idea (section 5.1) and prove, as an intermediate result, the convergence of the Gauss–Seidel iteration. In sections 5.2 and 5.3 we extend our theory to multiplicative Schwarz iterations. For our type of problem the Schwarz decomposition has to be somehow compatible with the graph of A.

In section 6 we consider a damped (relaxed) version of multiplicative Schwarz. It turns out that this variant converges for virtually any type of decomposition so that compatibility with the graph of A is not required.

Additive Schwarz iterations are studied in section 7. We will see that the theory developed in this paper can be directly applied to obtain new convergence results.

In section 8 we give an overview of other results and generalizations which can be obtained using our theory.

2. Definitions and auxiliary results. In this section we recall some definitions and preliminaries. If not stated otherwise, they can be found in [4]. The basic definitions concerning graphs are taken from [25].

2.1. Basics. Let $A = (a_{ij}) \in \mathbb{R}^{m\times n}$ be given. We assume that the reader is familiar with the concept of *nonnegativity* ($A \geq 0$) and positivity ($A > 0$); see, e.g., [4, 31].

For a given matrix $A \in \mathbb{R}^{n\times n}$ and sets $V_1, V_2 \subset \{1, \ldots, n\}$ the matrix $A[V_1, V_2]$ consists of a_{ij} satisfying $i \in V_1$ and $j \in V_2$. By $A[V_1]$ we denote the *principal minor* of A with respect to V_1.

For $A \in \mathbb{R}^{n\times n}$ we denote by $\mathcal{R}(A), \mathcal{N}(A), \sigma(A)$, and $\rho(A)$ the range, the null space, the spectrum, and the spectral radius of A, respectively. The term $\mathrm{ind}_\lambda(A)$ identifies the index of an eigenvalue λ of A. Finally, we define

$$\gamma(A) := \max\{|\lambda| : \lambda \in \sigma(A) \text{ and } \lambda \neq \rho(A)\}.$$

$A \in \mathbb{R}^{n\times n}$ is an *M-matrix* if $A = \beta I - B,\ B \geq 0$, and $\rho(B) \leq \beta$. If $\rho(B) = \beta$, the matrix A is singular; otherwise, it is nonsingular. A matrix $A \in \mathbb{R}^{n\times n}$ is a nonsingular M-matrix if and only if $A^{-1} \geq 0$. When A is a nonsingular M-matrix, each principal minor is a nonsingular M-matrix.

2.2. Graph theory. It is assumed that the basics from graph theory are known to the reader; see, e.g., [11].

Let $\Gamma_1 = (V, E_1)$ and $\Gamma_2 = (V, E_2)$ be directed graphs with identical vertex sets V. The union $\Gamma_1 \cup \Gamma_2$ is defined to be the union of the edge sets. The product graph $\Gamma_1\Gamma_2 = (V, E)$ is defined by $(u, w) \in E$ if there is a $v \in V$ such that $(u, v) \in E_1$ and $(v, w) \in E_2$. We write $\Gamma^2 = \Gamma\Gamma$ and $\Gamma^{k+1} = \Gamma^k\Gamma$. If V is a given vertex set, the graph $\Delta = (V, E)$ with $E := \{(v, v) : v \in V\}$ denotes the diagonal graph of V.

The *reflexive transitive closure* $\overline{\Gamma}$ of a graph Γ is defined to be

$$\overline{\Gamma} = \Delta \cup \Gamma \cup \Gamma^2 \cup \ldots.$$

Therefore, $u \in V$ has access to $w \in V$ in Γ if and only if (u, w) is an edge of $\overline{\Gamma}$.

For $A \in \mathbb{R}^{n\times n}$, we denote by $\Gamma(A) = (V(A), E(A))$ the corresponding directed graph. Now the vertices are also called *states* or *indices*.

LEMMA 2.1. *Let $A, B \in \mathbb{R}^{n\times n}$ be nonnegative, and let $\alpha \in \mathbb{R}$ be positive. Then*

$$\Gamma(\alpha A) = \Gamma(A), \quad \Gamma(A+B) = \Gamma(A) \cup \Gamma(B), \quad \textit{and} \quad \Gamma(AB) = \Gamma(A)\Gamma(B).$$

If A is a nonsingular M-matrix, then $\Gamma(A^{-1}) = \overline{\Gamma(A)}$.

2.3. Classification of vertices and eigenvectors. Let a graph Γ be given. Equivalence classes of Γ are always to be understood with respect to the communication relation. If $A \in \mathbb{R}^{n\times n}$ is some matrix, then A is called *irreducible* if $\Gamma(A)$ is strongly connected; i.e., there is only one equivalence class. Otherwise, A is called *reducible*.

THEOREM 2.2. *Let $A \in \mathbb{R}^{n\times n}$ be an irreducible M-matrix.*

(1) *If A is nonsingular, then $A^{-1} > 0$.*

(2) *If A is singular, then each proper principal minor of A is a nonsingular M-matrix.*

For an $A \in \mathbb{R}^{n\times n}$, the *classes* of A are the equivalence classes of $\Gamma(A)$. A class is called *final* if none of its vertices has access to another class. A class α is called *basic* if $\rho(A[\alpha]) = \rho(A)$; otherwise, α is called *nonbasic*.

We will need the following result on the relation between graphs of nonnegative matrices, positive eigenvectors, and their index. A proof can be found in [24].

THEOREM 2.3. *Let $B \in \mathbb{R}^{n\times n}$ be nonnegative, and let $\mathcal{N}((\rho(B)I - B)^k)$ with $k = \operatorname{ind}_{\rho(B)}(B)$ be the* algebraic eigenspace. *Assume that B has m basic classes $\alpha_1, \ldots, \alpha_m$. Then the algebraic eigenspace corresponding to $\rho(B)$ contains nonnegative vectors $v^{(1)}, \ldots, v^{(m)}$, such that $v_j^{(i)} > 0$ if and only if the index j has access to α_i in $\Gamma(B)$. Furthermore, any such collection is a basis of the algebraic eigenspace.*

THEOREM 2.4. *Let $B \in \mathbb{R}^{n\times n}$ be nonnegative; then there is a positive eigenvector z corresponding to the spectral radius if and only if the basic classes of B are exactly its final ones.*

2.4. Semiconvergence. For a given $T \in \mathbb{R}^{n\times n}$ we say that T is *semiconvergent* if $\lim_{k\longrightarrow\infty} T^k = T^*$ exists and, additionally, $T^* \neq 0$. This concept is used to analyze stationary iterations

$$x^{k+1} = Tx^k + c, \qquad k = 0, 1, 2, \ldots, \tag{2.1}$$

which are known to converge for every initial guess x^0 if $c \in \mathcal{R}(I - T)$ and T is semiconvergent.

A square matrix T is semiconvergent if and only if T satisfies the following three conditions (see [4, Lemma 7.6.9]):

$$(1)\ 1 \in \sigma(T), \quad (2)\ \gamma(T) < 1, \quad \text{and} \quad (3)\ \mathcal{R}(I-T) \oplus \mathcal{N}(I-T) = \mathbb{R}^n.$$

Furthermore, $T^* = \lim_{k\longrightarrow\infty} T^k$ is a projection onto $\mathcal{N}(I - T)$ along $\mathcal{R}(I - T)$.

Note that condition (3) states that there are no generalized eigenvectors to the eigenvalue 1. This is equivalent to $\operatorname{ind}_1(T) = 1$ and automatically fulfilled if $Tz = z$ for some positive vector z; see [4, 31]. If $T \geq 0$, then condition (2) can be replaced by T having all its diagonal elements positive [1] or, in the case that T is irreducible, it suffices to have one positive diagonal element; see [4, Corollary 2.2.28].

3. The model problem *MP*. In this section we present our model problem which represents the class of singular systems which we consider. We will see that the model problem exposes a nice graph structure which allows us to construct convergent Schwarz iterations.

Assume that for $A \in \mathbb{R}^{n\times n}$ there holds

$$A = I - B, \quad B \geq 0, \quad \rho(B) = 1. \tag{3.1}$$

For a given $b \in \mathcal{R}(A)$, the problem to be solved is to find a solution $x^* \in \mathbb{R}^n$ of

$$Ax = b \Leftrightarrow x = Bx + b \tag{3.2}$$

for $x \in \mathbb{R}^n$.

Equation (3.2) will not be considered in its greatest generality; only in the following case.

DEFINITION 3.1. *The model problem* MP *is to find a solution* x^* *of* (3.2) *satisfying* (3.1) *in the case that* $\mathcal{N}(A) = \operatorname{span}\{z\}$ *for a positive vector* z.

We now characterize $\mathcal{N}(A) = \operatorname{span}\{z\}$ with $z > 0$ via the graph structure of A. We say that a directed graph $\Gamma = (V, E)$ contains a directed *spanning tree* $\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T})$ (see, e.g., [9]) if

$$(1)\ \mathcal{T} \text{ is a directed tree}, \quad (2)\ V_\mathcal{T} = V, \quad \text{and} \quad (3)\ E_\mathcal{T} \subset E.$$

As we discuss only directed graphs, the term "directed" will be omitted in the rest of the paper.

DEFINITION 3.2. *Let* $B \in \mathbb{R}^{n\times n}$ *be such that* $B \geq 0$, $\rho(B) = 1$, *and let* $\Gamma(B) = (V(B), E(B))$ *be the corresponding graph. Then* B *is said to be an* ST-matrix *(ST for "spanning tree") if the following hold:*

(1) $\Gamma(B^T)$ *contains a spanning tree* $\mathcal{T}_B$,

(2) *if the index* $i_0 \in V(B)$ *is the root of* $\mathcal{T}_B$, *then* i_0 *has access to some* $j_0 \in V(B)$ *via* $(i_0, j_0) \in \Gamma(B)$, *and*

(3) *each class of* B *is final if and only if it is basic.*

The index j_0 *defined above will be called the* guard index.

Remark 3.1. Note that it might happen that the root i_0 communicates only with itself via $(i_0, i_0) \in \Gamma(B)$. As this leads to some problems when using Schwarz iterations, we will always assume $j_0 \neq i_0$ if not stated otherwise.

The following lemma characterizes ST-matrices. The results can directly be deduced from Theorems 2.3 and 2.4 and the theory of Markov chains [4, 26].

LEMMA 3.3. *Let* B *be an ST-matrix; then there is a permutation matrix* Π *such that*

$$\Pi B \Pi^T = \begin{pmatrix} D & 0 \\ E & F \end{pmatrix}. \tag{3.3}$$

Furthermore, the following hold:

(1) D *is square and irreducible.*

(2) *If* B *is irreducible, then* $\Pi = I$ *and* $B = D$.

(3) *If* i_0 *is the root index of any tree in* $\Gamma(B^T)$, *then* i_0 *resides in the index set belonging to* D.

(4) $\rho(B) = \rho(D) = 1$.

(5) $\rho(F) < 1$.

(6) *There exists a vector* $z > 0$ *such that* $Bz = z$, $\mathcal{N}(I - B) = \operatorname{span}\{z\}$, *and* $\operatorname{ind}_1(B) = 1$.

Remark 3.2. Note that the representation (3.3) can be efficiently calculated with the algorithm of Tarjan [29].

Now we state a few corollaries of Lemma 3.3.

COROLLARY 3.4. *Let $B \in \mathbb{R}^{n\times n}$ be an ST-matrix and let α be the set of indices of the final class of B. Then $\Gamma(B^T)$ contains at least $|\alpha|$ spanning trees with corresponding guard indices; i.e., each index in α might act as a root.*

Proof. It is easy to see that every index $i_0 \in \alpha$ has access to all other indices in $\Gamma(B^T)$. ∎

COROLLARY 3.5. *Let $B \in \mathbb{R}^{n\times n}$ be an irreducible ST-matrix. Then $\Gamma(B^T)$ contains at least n spanning trees with corresponding guard indices; i.e., every index can act as a root.*

COROLLARY 3.6. *If $B \in \mathbb{R}^{n\times n}$ is a symmetric ST-matrix, then B is irreducible.*

COROLLARY 3.7. *Consider a nonnegative matrix $B \in \mathbb{R}^{n\times n}$, $\rho(B) = 1$, and a positive vector $z > 0$ such that $Bz = z$. If B contains a spanning tree and a guard index, then B is an ST-matrix.*

If the final classes are exactly the basic ones and there is only one such class, then the existence of a spanning tree is not only sufficient but also necessary.

THEOREM 3.8. *Let $B \in \mathbb{R}^{n\times n}$ be nonnegative, and let $\rho(B) = 1$. Then B is an ST-matrix if and only if the final classes of B are exactly its basic ones and there is only one such class.*

Proof. The sufficiency is given in Lemma 3.3. The necessity is shown now. Assume that B has exactly one final class which is also the only basic class.

There is a permutation matrix Π such that (3.3) holds; then

$$C^T = \begin{pmatrix} D^T & E^T \\ 0 & F^T \end{pmatrix} \tag{3.4}$$

and D is irreducible, satisfying $\rho(D) = 1$ and $\rho(F) < 1$. Additionally, there exists a positive vector $z > 0$ such that $Bz = z$ (cf. Theorem 2.4). Let the index set $\{1, \dots, n\}$ be split with respect to (3.4) into sets V_1 and V_2, where V_1 corresponds to the indices of D. It remains to show that each index in V_2 is accessible from V_1 in $\Gamma(B^T)$. If this is proven, the existence of a spanning tree is obvious because any index belonging to D then has access to every other index in $\Gamma(B^T)$ (cf. Corollary 3.4).

Assume that there is a nonempty subset W_2 of V_2 containing all the indices that are not accessible from V_1. Then there is another permutation matrix $\tilde{\Pi}$ acting on V_2 such that

$$\tilde{\Pi}C\tilde{\Pi}^T = \begin{pmatrix} D & 0 & 0 \\ E_1 & F_{11} & F_{12} \\ E_2 & F_{21} & F_{22} \end{pmatrix} \text{ and } \tilde{\Pi}^T C^T \tilde{\Pi} = \begin{pmatrix} D^T & E_1^T & E_2^T \\ 0 & F_{11}^T & F_{21}^T \\ 0 & F_{12}^T & F_{22}^T \end{pmatrix}. \tag{3.5}$$

Here F_{22} corresponds to the index set $W_2 \subset V_2$ of all nonaccessible indices, while F_{11} corresponds to the indices $W_1 = V_2 \setminus W_2$.

Since each $j \in W_2$ is not accessible from V_1, one gets $E_2^T = 0$, whereas $E_1^T \neq 0$. But then $F_{21}^T = 0$, since all indices in W_1 are accessible from V_1; hence $F_{21}^T \neq 0$ would imply that an index $i \in V_1$ has access to some $j \in W_2$ via a $k \in W_1$. Thus

$$\tilde{\Pi}C\tilde{\Pi}^T = \begin{pmatrix} D & 0 & 0 \\ E_1 & F_{11} & F_{12} \\ 0 & 0 & F_{22} \end{pmatrix}.$$

If z is split into (z_1, z_2, z_3) with respect to V_1, W_1, and W_2, then $F_{22}z_3 = z_3 > 0$. Consequently, F_{22} is a final and basic class, contradicting the assumptions. As there is only one final and basic class, z is a basis for $\mathcal{N}(A)$. ∎

In order to give an alternative characterization of our model problem, we define the following class of singular M-matrices.

DEFINITION 3.9. *A matrix $A \in \mathbb{R}^{n\times n}$ is called an* STM-matrix *(STM for "spanning tree monotone") if $A = I - B$ and B is an ST-matrix.*

An STM-matrix naturally fulfills the requirements of the model problem. On the other hand, each matrix $A = I - B$ having a positive vector $z > 0$ such that $Az = 0$ and $\dim \mathcal{N}(A) = 1$ is an STM-matrix by Theorem 3.8. Thus, the model problem can be restated as follows.

Given an STM-matrix $A \in \mathbb{R}^{n\times n}$ and $b \in \mathcal{R}(A)$, find a solution $x^* \in \mathbb{R}^n$ of

$$Ax = b,\ x \in \mathbb{R}^n.$$

4. Schwarz methods. We give a brief introduction to algebraic Schwarz methods following, e.g., [3, 8, 13, 16, 19, 20]. For technical reasons we will distinguish between partitionings and decompositions.

Let the finite set $S = \{1, \dots, n\}$ be given. The nonempty sets $S_1, \dots, S_p$ are a *decomposition* of S if

$$\bigcup_{i=1}^{p} S_i = S.$$

If an index $j \in S$ appears in more than one set, we speak of overlap. The measure of overlap is the maximum number of sets any index $j \in S$ belongs to, i.e.,

$$q = \max_{j=1,\dots,n} |\{i : j \in S_i\}|. \tag{4.1}$$

We have $q = 1$ if and only if no overlap occurs; see, e.g., [3]. In the latter case we say that $S_1, \dots, S_p$ form a *partitioning*.

If $A \in \mathbb{R}^{n\times n}$, then a partitioning or a decomposition $S_1, \dots, S_p$ of $\{1, \dots, n\}$ is called *regular* (with respect to A) if $A[S_i]$ is invertible for every $i = 1, \dots, p$. Note that it is impossible to find a regular decomposition of an STM-matrix if the basic class consists only of one index.

For a given $A \in \mathbb{R}^{n\times n}$ and a regular decomposition $S_1, \dots, S_p$ we define restriction operators $R_i \in \mathbb{R}^{|S_i|\times n}$ by the rows of the identity matrix corresponding to the indices in S_i; e.g., if $n = 6$ and $S_1 = \{1, 3, 2\}$, then

$$R_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

With the above restriction operators, a projection onto the subspace associated with S_i is given by

$$P_i := R_i^T (R_i A R_i^T)^{-1} R_i A. \tag{4.2}$$

A *(one level) multiplicative Schwarz method* for an initial vector x^0 is given by the stationary iteration (2.1) with

$$T = T_\mu := (I - P_1) \cdot (I - P_2) \cdots (I - P_p). \tag{4.3}$$

The *(damped) additive Schwarz method* is also defined by (2.1), but now

$$T = T_\theta := I - \theta \sum_{i=1}^{p} P_i = I - \theta \sum_{i=1}^{p} R_i^T (R_i A R_i^T)^{-1} R_i A \tag{4.4}$$

and $\theta \in (0, 1)$ is a damping factor.

Note that the usage of partitionings instead of decompositions puts (4.3) into a block Gauß–Seidel and (4.4) into a damped block Jacobi operator.

The following proposition is easy to prove and summarizes the properties of the operators defined above. The proof for the semiconvergence of T_θ can be found in [8]. The rest is easy to obtain.

PROPOSITION 4.1. *Let $A \in \mathbb{R}^{n\times n}$ be an STM-matrix and let $S_1, \dots, S_p$ be a regular decomposition of $\{1, \dots, n\}$. Let $z \in \mathbb{R}^n$ be the positive vector satisfying $Az = 0$. Then, with operators as given in* (4.2) *and* (4.3)*:*

(1) $I - P_i \geq 0,\ i = 1, \dots, p,$
(2) $(I - P_i)z = z,\ i = 1, \dots, p,$
(3) $\rho(I - P_i) = 1,\ i = 1, \dots, p,$
(4) $\mathrm{ind}_1(I - P_i) = 1,\ i = 1, \dots, p,$
(5) $T_\mu \geq 0,$
(6) $T_\mu z = z,$
(7) $\rho(T_\mu) = 1,$
(8) $\mathrm{ind}_1(T_\mu) = 1.$

If T_θ is defined by (4.4) *and $\theta \in (0, 1/q)$, where q is the measure of overlap, then assertions* (5)–(8) *apply verbatim to T_θ. Moreover, T_θ has all its diagonal elements positive, and therefore T_θ is semiconvergent.*

Comparing Proposition 4.1 with the conditions of semiconvergence stated in section 2.4, we see that they match directly except for multiplicative Schwarz. But a closer look on the iterations reveals that, for both additive and multiplicative Schwarz, consistency is nontrivial; i.e., it is not clear that

$$\mathcal{N}(I - T_\mu) = \mathcal{N}(I - T_\theta) = \mathcal{N}(A) \tag{4.5}$$

holds for an STM-matrix A.

Indeed, consider the STM-matrix

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1/2 & 0 & 1 & 0 & -1/2 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix},$$

and put $S_1 = \{1, 2, 3\}, S_2 = \{4, 5, 7\}$, and $S_3 = \{6, 7\}$. Then

$$T_\mu := (I - P_1)(I - P_2)(I - P_3) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

This is a rank two semiconvergent matrix. Thus

$$\mathcal{N}(I - T_\mu) = \mathrm{span}\{(1, \dots, 1)^T, e_6\} \neq \mathcal{N}(A) = \mathrm{span}\{(1, \dots, 1)^T\},$$

and the iteration might fail to converge against a desired solution. In what follows, we therefore have to investigate consistency and semiconvergence for multiplicative Schwarz, whereas for additive Schwarz we have only to consider consistency.

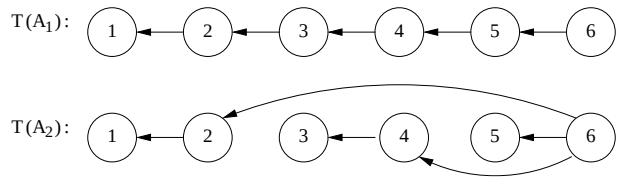


FIG. 5.1. *Trees of A_1 and A_2.*

Note that a consistency problem cannot occur if we use partitions instead of decompositions, i.e., we use block Jacobi or block Gauß–Seidel iterations. This is not hard to see.

5. Convergence of multiplicative Schwarz for *MP*. In this section we define decompositions for an STM-matrix which allows us to prove that a corresponding operator of the form (4.3) is semiconvergent and consistent. As the basic ideas can be derived from simple Gauß–Seidel iterations, we will start with partitionings consisting of singletons.

5.1. The basic idea. Consider the following STM-matrices $A_i = I - B_i,\ B_i \geq 0$ (we show only the nonzero entries):

$$A_1 = \begin{pmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & 1 & -1 & & \\ & & & 1 & -1 & \\ & & & & 1 & -1 \\ -1 & & & & & 1 \end{pmatrix}, A_2 = \begin{pmatrix} 1 & -1 & & & & \\ & 1 & & & & -1 \\ & & 1 & -1 & & \\ & & & 1 & & -1 \\ & & & & 1 & -1 \\ -1 & & & & & 1 \end{pmatrix}.$$

We assume a multiplicative Schwarz iteration for the partitioning $S_1, \ldots, S_6$, where $S_i = \{i\}$; i.e., we consider a Gauß–Seidel iteration. Then the projection $I - P_3$, e.g., for A_2, becomes

$$I - P_3 = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 0 & 1 & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{pmatrix}.$$

It is easy to see that, for both matrices A_1 and A_2, the operator

$$T_\mu = (I - P_1) \cdots (I - P_6)$$

has a positive first column. And this can be explained by considering the spanning trees of the respective matrices (see Figure 5.1) and multiplying the B_i by an appropriate vector: Consider a vector $x = (\xi, 0, 0, 0, 0, 0)^T, \xi > 0$, and the product $B_i x$ for some B_i. Then the initial value ξ is stored in state 6, the root. Another application of B_i to $B_i x$ reveals that ξ is carried to the states which are direct children of the root, and so on. Hence ξ "flows" through the tree, until it reaches the leaves. The product operator T_μ combines this transport into a single step; i.e., it can be interpreted as introducing a shortcut. The most important property of T_μ is given in the following theorem.

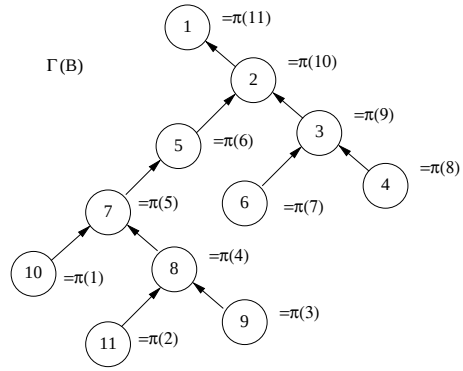


FIG. 5.2. *A flow compatible numbering.*

THEOREM 5.1. *Let $B \in \mathbb{R}^{n\times n}$ be nonnegative with $\rho(B) = 1$. Assume that B has a positive column and there exists $z > 0$ such that $Bz = z$. Then $\gamma(B) < 1$; i.e., B is semiconvergent and $\mathcal{N}(I - B) = \operatorname{span}\{z\}$.*

The above result has been used and proved in [15] and [26]. But it is easy to prove it in the ST-matrix context.

Proof. Let column i_0 of B be positive. Then B contains a spanning tree of height one with the root index i_0 (it is exactly the column i_0 and here we allow the root to coincide with the guard index; cf. Remark 3.1). Since $Bz = z$ for a some z, it follows from Corollary 3.7 that B is an ST-matrix and from section 2.4 that $\rho(B) = 1$ and $\operatorname{ind}_1 B = 1$. By Lemma 3.3, there is a permutation matrix Π such that (3.3) holds, where $\rho(F) < 1$, $\rho(D) = 1$, and D is irreducible. The existence of a positive column is invariant under symmetric permutation. Hence D has a positive diagonal element. But then, $\gamma(D) < 1$ from section 2.4. Thus $\gamma(B) < 1$ and the semiconvergence follows. ∎

Theorem 5.1 gives us the direction we need. If we can characterize decompositions which lead us to operators having a positive column, then the iteration is both semiconvergent and consistent. The example above gives us a hint how to obtain such operators—simply by respecting the above-mentioned "flow." As we will see, the "flow" can be generalized to decompositions and is the key concept behind our investigations.

Based on this idea, we are looking for decompositions such that we have a mapping from the set of STM-matrices (original matrix) into the set of consistent semiconvergent ST-matrices (operator of the multiplicative Schwarz iteration).

DEFINITION 5.2. *Let A be an STM-matrix (or ST-matrix), and let $\mathcal{T}$ be an arbitrary spanning tree in $\Gamma(A^T)$ with some guard index. A* flow compatible numbering (or permutation) *of the vertices of $\mathcal{T}$ is a permutation $\pi : \{1,\dots,n\} \longrightarrow \{1,\dots,n\}$ such that if there is a path from $\pi(i)$ to $\pi(j)$ in $\mathcal{T}$, then $i > j$ for each $1 \le i, j \le n$, $i \ne j$. Any such permutation is also called* flow compatible *(with respect to $\mathcal{T}$).*

Figure 5.2 shows a flow compatible numbering for an ST-matrix B (guard edges have been left out). The example defines a permutation $(10, 11, 9, 8, 7, 5, 6, 4, 3, 2, 1)$, but note that the permutation $(9, 11, 8, 4, 6, 3, 10, 7, 5, 2, 1)$ is also flow compatible.

Remark 5.1. Note that a flow compatible numbering can easily be calculated using a simple depth first search strategy [9] if we start within the strongly connected component (cf. Remark 3.2 and [5, section 4.3]).

We now have the following theorem, which also gives a proof for the classical Gauß–Seidel iteration for STM-matrices.

THEOREM 5.3. *Let $A = I - B \in \mathbb{R}^{n\times n}$ be an STM-matrix, and let π be a flow compatible permutation with respect to some tree $\mathcal{T} \subset \Gamma(B^T)$. Assume that the regular partitioning $S_1, \dots, S_n$ is such that $S_i = \{i\}$, and let P_i be the corresponding projections given by* (4.3) *for $i = 1, \dots, n$. Then*

$$(I - P_{\pi(1)}) \cdot (I - P_{\pi(2)}) \cdots (I - P_{\pi(n)})$$

has at least one positive column and is therefore semiconvergent.

Proof. We prove the theorem using a straightforward calculation of the matrix product.

Since the node $\pi(n)$ is always the root of a tree $\mathcal{T}$ in $\Gamma(B^T)$ (cf. Definition 5.2), there is a guard index j_0 such that $(I - P_{\pi(n)})_{\pi(n),j_0} > 0$. It will first be shown that

$$((I - P_{\pi(k)}) \cdot (I - P_{\pi(k+1)}) \cdots (I - P_{\pi(n)}))_{\pi(k),j_0} > 0$$

for an arbitrary $k \in \{1, \dots, n\}$. As the proposition is obvious for $k = n$, let $1 \le k < n$ be arbitrary but fixed.

Case 1. $(\pi(k), \pi(k+1)) \in \Gamma(B)$. In this case, the node $\pi(k+1)$ represents the parent of $\pi(k)$ in $\mathcal{T}$. Thus we have $(I - P_{\pi(k)})_{\pi(k),\pi(k+1)} > 0$, and there is an index $l \in \mathbb{N},\ k < l \le n$, such that $(I - P_{\pi(k+1)})_{\pi(k+1),\pi(l)} > 0$. The latter holds, since there is a path from $\pi(k+1)$ to $\pi(n)$ in $\mathcal{T}$. But then

$$((I - P_{\pi(k)}) \cdot (I - P_{\pi(k+1)}))_{\pi(k),\pi(l)} > 0.$$

Case 2. $(\pi(k), \pi(k+1)) \notin \Gamma(B)$. In this case, the index $\pi(k)$ must have access to $\pi(l)$ for some $l \in \mathbb{N},\ k+1 < l \le n$, since there is again a path from $\pi(k)$ to $\pi(n)$ in $\mathcal{T}$. The construction of the operators implies that $(I - P_{\pi(k+1)})_{\pi(l),\pi(l)} > 0$ because $\pi(l) \notin S_{k+1}$; thus

$$((I - P_{\pi(k)}) \cdot (I - P_{\pi(k+1)}))_{\pi(k),\pi(l)} > 0.$$

A simple induction leads to

$$((I - P_{\pi(k)}) \cdots (I - P_{\pi(n-1)}))_{\pi(k),\pi(l)} > 0$$

for some $l \in \mathbb{N},\ n-1 < l \le n$, i.e., $l = n$. But $(I - P_{\pi(n)})_{\pi(n),j_0} > 0$, and there holds

$$((I - P_{\pi(k)}) \cdots (I - P_{\pi(n)}))_{\pi(k),j_0} > 0.$$

To finish the proof of the theorem observe that $(I - P_{\pi(j)})_{\pi(k),\pi(k)} > 0$ follows for all $1 \le j < k$ from the construction of $(I - P_{\pi(j)})$. Hence

$$((I - P_{\pi(1)}) \cdot (I - P_{\pi(2)}) \cdots (I - P_{\pi(n)}))_{\pi(k),j_0} > 0.$$

This is the theorem since k was arbitrary. ∎

Theorem 5.3 gives us information about the order of row updates to be used in the Gauß–Seidel iteration. As the proof shows, if the matrix is full, any order works.

COROLLARY 5.4. *Let $A = I - B \in \mathbb{R}^{n\times n}$. If $B > 0$ and $\rho(B) = 1$, then the Gauß–Seidel iteration converges.*

5.2. Block operators. We now extend the concept of flow compatibility to block operators. We start with an investigation of the structure of a single projection.

The following lemma is fundamental, but we need additional notation. To this purpose, let A be an STM-matrix, and let $\mathcal{T}$ be any spanning tree in $\Gamma(A^T)$.

We say a vertex i has *access to a vertex* j *along* $\mathcal{T}$ if there is a path $(j = l_1, l_2, \ldots, l_k = i)$ in $\mathcal{T}$. This is denoted by $i \to_{\mathcal{T}} j$.

Remark 5.2.

(1) If $i \to_{\mathcal{T}} j$, then $(i = l_k, l_{k-1}, \ldots, l_1 = j)$ is a path in $\Gamma(A)$.

(2) The access relation *along* $\mathcal{T}$ is useful, because $\mathcal{T}$ represents a minimal structure of positive elements needed to determine for which decompositions multiplicative Schwarz iterations converge. All other positive elements of A can be ignored.

LEMMA 5.5. *Let A be an STM-matrix, and let $\mathcal{T}$ be an arbitrary spanning tree in $\Gamma(A^T)$. Let π be a flow compatible permutation corresponding to $\mathcal{T}$. Let the tuple*

$$V := (\pi(k), \pi(k+1), \ldots, \pi(l)), \quad 1 \le k < l \le n, \; l - k < n - 1,$$

be chosen in such a way that $A[V]^{-1}$ exists and denote by Π the permutation matrix corresponding to π. According to (4.2)*, let the matrix $I - P$ be defined through*

$$\Pi(I-P)\Pi^T = \begin{pmatrix} I & 0 & 0 \\ M^{-1}N_L & 0 & M^{-1}N_R \\ 0 & 0 & I \end{pmatrix}, \tag{5.1}$$

where $N_L = -A[V, (\pi(1), \ldots, \pi(k-1))]$, $N_R = -A[V, (\pi(l+1), \ldots, \pi(n))]$, and $M = A[V]$. Then for each index j_0, $k \le j_0 \le l$, one of the following three conditions holds:

(1) *If $j_0 = n$, then there exists a node $\pi(i_0) \notin V$ such that $\pi(n) \to \pi(i_0)$ in $\Gamma(A)$ and $(\pi(n), \pi(i_0)) \in \Gamma(I - P)$.*

(2) *If $j_0 \neq n$ and $(\pi(i_0), \pi(j_0)) \in \mathcal{T}$ for some node $\pi(i_0) \notin V$, then $(\pi(j_0), \pi(i_0)) \in \Gamma(I - P)$.*

(3) *If $j_0 \neq n$ and $(\pi(h), \pi(j_0)) \in \mathcal{T}$ for some node $\pi(h) \in V$, then there exists $\pi(i_0) \notin V$ such that $\pi(j_0) \to_{\mathcal{T}} \pi(i_0)$. For any such $\pi(i_0)$ we have $(\pi(j_0), \pi(i_0)) \in \Gamma(I - P)$.*

Furthermore, the node $\pi(i_0)$ in (1)*,* (2)*, and* (3) *satisfies the following:*

(4) *If $l < n$, then $i_0 > l$.*

(5) *If $l = n$, then $i_0 < k$ and i_0 is given by assertion* (1) *(with $j_0 = l$).*

In case (4) *i_0 is unique (with respect to $\mathcal{T}$). In case* (5) *i_0 depends on the chosen guard.*

Before we prove the lemma, we provide a few explanations.

Remark 5.3. The lemma has an easy interpretation if we consider the representation (5.1).

(i) The case $\pi(n) \in V$ plays a special role. Assertion (1) states that there exists a path in $\Gamma(A)$ starting at $\pi(n)$, which leads out of V, and this path is replaced by an edge in $\Gamma(I - P)$, which resides in $M^{-1}N_L$ (note that N_R is empty).

(ii) Assertion (2) says that if $\pi(j_0) \in V$ has direct access to $\pi(i_0) \notin V$ along $\mathcal{T}$ (i.e., $(\pi(j_0), \pi(i_0)) \in \Gamma(A)$), then this edge is still in $\Gamma(I-P)$; i.e., connections leading out of V are preserved.

(iii) Assertion (3) says that if $\pi(j_0) \in V$ has direct access to some element in V along $\mathcal{T}$, then there is a path in V along $\mathcal{T}$ leading to some $\pi(i_0) \notin V$. Furthermore, for each such path there will be an edge in $\Gamma(I - P)$. We can interpret this as that a multiplication with M^{-1} introduces a shortcut in $\mathcal{T}$, with the flow itself being preserved.

(iv) Assertion (4) states that the edges of interest lie in $M^{-1}N_R$ except in the case in which the root is in V, which is considered in assertion (5). Then, since each vertex has access to the root, access to some vertex outside V is guaranteed by assertion (1).

Figure 5.3 describes the situation for an STM-matrix $A = I - B$. The figure shows the graph of A and the tuple $V = (\pi(4), \pi(5), \pi(6), \pi(7)) = (8, 7, 5, 6)$. The graph $\Gamma(I - P)$ is the graph of P given by (5.1) with respect to V. Note that the dashed edges in $\Gamma(I - P)$ are shortcuts of paths existing in $\Gamma(A)$. The diagonal edges and the guard edges in $\Gamma(A)$ are not shown.

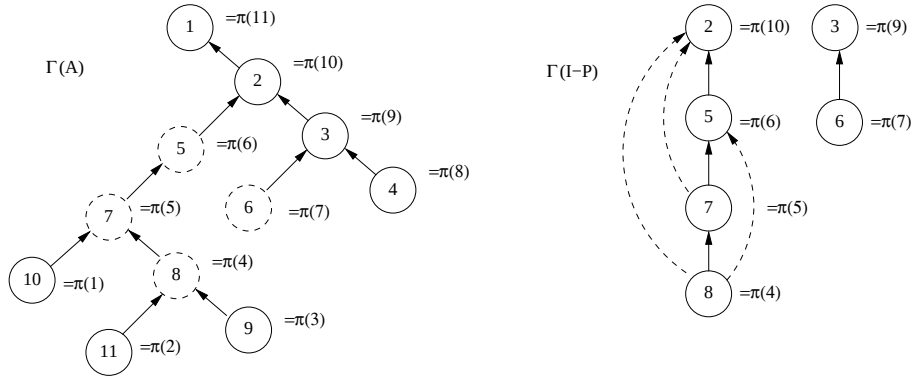


FIG. 5.3. *Graph of a matrix and of a corresponding projection.*

Proof. The proof of Lemma 5.5 is a bit technical and is therefore split into several parts.

Part 1: *Classification of vertices in* V. Define boundary, inner, and outer vertices $B(V), I(V), O(V)$, respectively, by

$$\begin{aligned} B(V) &:= \{v \in V : \text{ there exist } w \notin V \text{ such that } (w, v) \in \mathcal{T}\}, \\ I(V) &:= \{v \in V : \text{ there exist } u \in V \text{ such that } (u, v) \in \mathcal{T}\}, \\ O(V) &:= V \setminus (B(V) \cup I(V)). \end{aligned}$$

Taking the set V from Figure 5.3, we have $B(V) = \{\pi(6), \pi(7)\} = \{5, 6\}$, $I(V) = \{\pi(4), \pi(5)\} = \{8, 7\}$, and $O(V) = \emptyset$.

Now we have

(i) $B(V) \cap I(V) = \emptyset$,
(ii) $B(V) \cup O(V) \cup I(V) = V$,
(iii) $O(V) \neq \emptyset \Rightarrow O(V) = \{\pi(n)\}$, and
(iv) $B(V) = \emptyset \Leftrightarrow O(V) \neq \emptyset$.

To prove this let $v \in V$ be arbitrary. Assume there exists a w such that $(w, v) \in \mathcal{T}$. Then w is unique since $\mathcal{T}$ is tree. If $w \in V$, then $v \in I(V)$; else $v \in B(V)$. If there exists no w satisfying $(w, v) \in \mathcal{T}$, then $v = \pi(n)$ and obviously $\pi(n) \in O(V)$. Since the root is unique, this proves (i)–(iii).

Suppose $B(V) = \emptyset$. Every vertex, except the root, has a parent, but no vertex $v \in V$ has an adjacent vertex $w \notin V$ in $\mathcal{T}$. Thus, every vertex $v \in V$ has an adjacent vertex in V along $\mathcal{T}$ (except the root), and therefore $\pi(n) \in V$ holds necessarily.

Now assume $O(V) \neq \emptyset$; then $O(V) = \{\pi(n)\}$ by (iii) and $l = n$ by the flow compatibility and the definition of V. If there is a $v \in B(V)$, then there exists a

$w \notin V$ satisfying $(w, v) \in \mathcal{T}$. Additionally, there are numbers $k_1, k_2 \in \{1, \dots, n\}$ such that $v = \pi(k_1)$ and $w = \pi(k_2)$. Since π is flow compatible, $k \le k_1 < k_2 \le n$. But then $w = \pi(k_2) \in V$, which is a contradiction, and (iv) follows.

Part 2: *The case* $\pi(n) \in V$. Here we prove the existence of the index i_0 as stated in assertion (1) and (i) in Remark 5.3. We have

$$V := (\pi(k), \pi(k+1), \dots, \pi(n)), \quad 1 < k < n,$$

and $\pi(j_0) = \pi(n)$. Now we show that there is always a path in $\Gamma(A[V]) = \Gamma(M)$ starting at $\pi(n)$ which leads us to some $\pi(i_0)$ not in V.

As A is an STM-matrix, there is a guard index $k_0 \in \{1, \dots, n\}$ such that $(A)_{\pi(j_0),\pi(k_0)} > 0$. If $\pi(k_0) \notin V$, we have found the path and choose $i_0 := k_0$.

Hence assume $\pi(k_0) \in V$. Let α be the final and basic class of A. Denote $\tilde{\alpha} = \pi(\alpha)$. Then there is an index $\pi(i_0) \in \tilde{\alpha}$ which fulfills $\pi(i_0) \notin V$. Otherwise, the strongly connected class $\tilde{\alpha}$ would be a subset of V and because $0 \in \sigma(A[\tilde{\alpha}])$ we also have $0 \in \sigma(A[V])$ (cf. Lemma 3.3). The latter is impossible because $M^{-1} = A[V]^{-1}$ exists by assumption.

Now there exist $\pi(i_0) \in \tilde{\alpha} \backslash V$ and a path $(\pi(n) = \pi(i_k), \dots, \pi(i_1), \pi(i_0))$ from $\pi(n)$ to $\pi(i_0)$ in $\Gamma(A)$. We can choose the path such that $(\pi(i_k), \dots, \pi(i_1)) \subset \Gamma(A[V]) = \Gamma(M)$. Then $(\pi(i_1), \pi(i_0)) \in \Gamma(A)$ and, by the construction, $\pi(i_0) \notin V$ and $i_0 < k < l = n$; i.e., we have found a path leading outside V.

Additionally, by Lemma 2.1, $\Gamma(M^{-1}) = \overline{\Gamma(M)}$; thus there is an edge $(\pi(n), \pi(i_1)) \in \Gamma(M^{-1})$. We will see that this part of the path will be cut short in the final operator (we give an example of this situation later).

Part 3: *Inner vertices.* Now we analyze inner vertices as mentioned in (iii) in Remark 5.3. Let $v \in I(V)$; then there exists exactly one vertex $w \in V$ such that $(w, v) \in \mathcal{T}$. If $w \in I(V)$, we continue inductively until a vertex $u \notin I(V)$. Then $u \in B(V)$ or $u = \pi(n)$ (part 1, (iv)). In either case there is a unique path $p = (u = \pi(i_k), \pi(i_{k-1}), \dots, \pi(i_1) = v)$ in $\mathcal{T}$; i.e., $v \to_{\mathcal{T}} u$. Furthermore, $l \ge i_k > i_{k-1} > \cdots > i_1 \ge k$ and $p^T = (\pi(i_1), \dots, \pi(i_k))$ is a path in $\Gamma(M)$. As in part 2, we have $\Gamma(M^{-1}) = \overline{\Gamma(M)}$, i.e., $(v, u) \in \Gamma(M^{-1})$. So the path in $\mathcal{T}$ will be cut short in $\Gamma(M^{-1})$ by the vertex $(v, u) = (\pi(i_1), \pi(i_k))$. Additionally, we have $i_k > i_1$.

Part 4: *Proving the assertions.* Now we use the results from the previous parts to prove the assertions for an arbitrary but fixed j_0 with $k \le j_0 \le l$. Write $I - P$ as follows:

$$\Pi(I - P)\Pi^T = \begin{pmatrix} I & 0 & 0 \\ 0 & M^{-1} & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ N_L & 0 & N_R \\ 0 & 0 & I \end{pmatrix} =: M_n^{-1} N_n.$$

By Lemma 2.1

$$\Gamma(M_n^{-1} N_n) = \Gamma(M_n^{-1})\Gamma(N_n) = \overline{\Gamma(M_n)}\Gamma(N_n). \tag{5.2}$$

Case (a) $\pi(j_0) \in B(V)$. There exists exactly one $\pi(i_0)$ satisfying $(\pi(j_0), \pi(i_0))_{\mathcal{T}}$ and $\pi(i_0) \notin V$. But then $(\pi(j_0), \pi(i_0)) \in \Gamma(N_R)$ since $i_0 > j_0$ by the flow compatibility.

With $\Delta \subset \overline{\Gamma(M_n)}$ and (5.2) the relation $(\pi(j_0), \pi(i_0)) \in \Gamma(I - P)$ follows. Hence, assertion (2) combined with (4) is proven.

Case (b) $\pi(j_0) \in I(V)$ *and* $B(V) \neq \emptyset$. In view of part 3 there exists exactly one $\pi(h) \in B(V)$ such that $(\pi(j_0), \pi(h)) \in \overline{\Gamma(M_n)}$ and $h > j_0$. Applying case (a) to h instead of j_0, we see that there is exactly one edge $(\pi(h), \pi(i_0)) \in \Gamma(N_R)$. From (5.2) we again get $(\pi(j_0), \pi(i_0)) \in \Gamma(I-P)$, and $i_0 > h > j_0$. This is assertion (3) combined with (4).

Case (c) $\pi(j_0) = \pi(n)$. By part 2 there are vertices $\pi(i_0) \notin V$ and $\pi(j_1) \in V$ such that $(\pi(j_1), \pi(i_0)) \in \Gamma(A)$ and $(\pi(j_0), \pi(j_1)) \in \Gamma(M^{-1})$. But $i_0 < k < l = n$; thus $(\pi(j_1), \pi(i_0)) \in \Gamma(N_L)$, and again by (5.2), $(\pi(j_0), \pi(i_0)) \in \Gamma(I-P)$. This is assertion (1) combined with (5).

Case (d) $\pi(j_0) \in I(V)$ *and* $\pi(n) \in V$. By part 1, $B(V) = \emptyset$. Part 3 implies that there exists an edge $(\pi(j_0), \pi(n)) \in \overline{\Gamma(M_n)}$ which is unique in $\mathcal{T}$. Applying case (c) shows the existence of edges $(\pi(j_1), \pi(i_0)) \in \Gamma(N_L)$ and $(\pi(j_0), \pi(j_1)) \in \overline{\Gamma(M_n)}$. This implies $(\pi(j_0), \pi(i_0)) \in \Gamma(I-P)$ and $j_0 \geq k > i_0$. Hence, assertion (2) combined with (5) is proven.

This finishes the proof of the lemma. ∎

Remark 5.4. The lemma also holds for the projection P instead of $I - P$.

The proof shows that Lemma 5.5 holds independently of the order within V; hence, V can be interpreted as a set rather than a tuple.

DEFINITION 5.6. *Let $A \in \mathbb{R}^{n\times n}$ be an STM-matrix, and let π be a flow compatible numbering of a spanning tree $\mathcal{T} \subset \Gamma(A^T)$. A regular partitioning $S_1, \ldots, S_p$ of $(\pi(1), \ldots, \pi(n))$ is termed a* block flow compatible *partitioning if for $1 \leq k < l \leq p$ we have*

$$\max\{j : \pi(j) \in S_k\} < \min\{j : \pi(j) \in S_l\}.$$

We will use Lemma 5.5 and Definition 5.6 to prove the convergence of multiplicative Schwarz iterations. Before we start with this, we consider some other applications of the results presented so far. The first application concerns the block Jacobi iteration operator and will be used to motivate the approach in section 6.

COROLLARY 5.7. *Let $A \in \mathbb{R}^{n\times n}$ be an STM-matrix, and let π be a flow compatible numbering of a spanning tree $\mathcal{T} \subset \Gamma(A^T)$. Consider a block flow compatible partitioning $S_1, \ldots, S_p$ and define $M_i := A[S_i]$, $M = \operatorname{diag}(M_1, \ldots, M_p)$, and $N = M - A$. Then $M^{-1}N$ is an ST-matrix.*

Proof. Since the partitioning is regular, each M_i is invertible. By Lemma 5.5, each index $u \in S_i, i < p$ has access to some $v \in S_j, j > i$, in $\Gamma(M^{-1}N)$. Therefore, each index $u \in \{1, \ldots, n\} \setminus S_p$ has access to some $v \in S_p$. Furthermore, each index $v \in S_p$ has access to a single index $w \notin S_p$, which is the guard index of $\pi(n) \in S_p$. Hence $\Gamma((M^{-1}N)^T)$ contains a spanning tree. Since $w \notin S_p$, we have $w \in S_i$ for some $i < p$ and w has access to some index in $S_j, j > i$. Hence there is a path from w to itself in $\Gamma(M^{-1}N)$. The vertex w can be considered as the root, and there must exist a guard index, since there is a path from w to itself. The ST-matrix property follows now from Corollary 3.7. ∎

COROLLARY 5.8. *With the assumptions of Corollary* 5.7 *there exists a spanning tree $\widetilde{\mathcal{T}}$ in $M^{-1}N$ of height at most p.*

To give an example to illustrate Corollary 5.7 and Lemma 5.5, consider the STM-matrix $A = I - B$, where the graph of B is given by Figure 5.3 and the guard index of the root index 1 is index 3. Then

$$A = \begin{pmatrix}
1 & & -1 & & & & & & & & \\
-1 & 1 & & & & & & & & & \\
 & -1 & 1 & & & & & -1 & & & \\
 & & -1 & 1 & & & & & & & \\
 & -1 & & & 1 & & & & & & \\
 & & -1 & & & 1 & & & & & \\
 & & & & -1 & & 1 & & & & \\
 & & & & & & -1 & 1 & & & \\
 & & & & & & -1 & & & 1 & \\
 & & & & & & & -1 & & & 1
\end{pmatrix}.$$

Applying the flow compatible permutation given in Figure 5.3, A becomes

$$\Pi A \Pi^T = \left(\begin{array}{ccc|ccc|ccc|ccc}
1 & & & & -1 & & & & & & & \\
 & 1 & & -1 & & & & & & & & \\
 & & 1 & -1 & & & & & & & & \\
\hline
 & & & 1 & -1 & & & & & & & \\
 & & & & 1 & -1 & & & & & & \\
 & & & & & 1 & & & & -1 & & \\
 & & & & & & 1 & & -1 & & & \\
\hline
 & & & & & & & 1 & -1 & & & \\
 & & & -1 & & & & & 1 & -1 & & \\
 & & & & & & & & & 1 & -1 & \\
 & & & & & & & & -1 & & & 1
\end{array}\right).$$

Now consider the (permuted) partition $S_1 = \{1, 2, 3\}, S_2 = \{4, 5, 6, 7\}$, and $S_3 = \{8, 9, 10, 11\}$. To give an example for the proof of Lemma 5.5 note that

(i) S_1 consists of only boundary nodes,

(ii) S_2 consists of both boundary and inner nodes,

(iii) S_3 has one outer node (the root) and some inner nodes, and

(iv) the guard index of the root lies within S_3; thus S_3 is an example as discussed in part 2 of the proof of Lemma 5.5.

To illustrate Corollary 5.7, the block Jacobi iteration matrix becomes

$$M^{-1}N = \left(\begin{array}{ccc|ccc|ccc|ccc}
 & & & & 1 & & & & & & & \\
 & & & 1 & & & & & & & & \\
 & & & 1 & & & & & & & & \\
\hline
 & & & & & & & & & 1 & & \\
 & & & & & & & & & 1 & & \\
 & & & & & & & & & 1 & & \\
 & & & & & & & & 1 & & & \\
\hline
 & & & 1 & & & & & & & & \\
 & & & 1 & & & & & & & & \\
 & & & 1 & & & & & & & & \\
 & & & 1 & & & & & & & &
\end{array}\right).$$

The matrix $M^{-1}N$ is obviously an ST-matrix with the root index $4 = \pi^{-1}(8)$ and the guard index $10 = \pi^{-1}(2)$. Note that $M^{-1}N$ is not semiconvergent.

Now let $(I - P_1), (I - P_2)$, and $(I - P_3)$ be the operators given by (5.1) with respect to $S_i, i = 1, 2, 3$. Then the nontrivial block row of $I - P_i$ corresponds to the ith block row of $M^{-1}N$. Additionally, a short computation shows that the product $(I - P_1) \cdot (I - P_2) \cdot (I - P_3)$ has a positive fourth column and is thus semiconvergent.

This is not very surprising because now the block rows behave like single states and the order of the projections in the product can be interpreted as a flow compatible numbering of a (block) spanning tree of height 3 (cf. Corollary 5.8). In the general case, the following result holds.

THEOREM 5.9. *Let $A \in \mathbb{R}^{n\times n}$ be an STM-matrix and $S_1, \dots, S_p$ be a block flow compatible partitioning. Let $(I - P_i)$ be given by* (5.1)*; then the product*

$$P := (I - P_1) \cdots (I - P_p)$$

has at least one positive column and is semiconvergent.

Proof. This theorem is a special case of Theorem 5.13 to be proven later. □

Theorem 5.9 has a corollary which has the same meaning as Corollary 5.4 has for Theorem 5.3.

COROLLARY 5.10. *Let $A = I - B \in \mathbb{R}^{n\times n}$. Assume $B > 0$ and $\rho(B) = 1$; then the block Gauß–Seidel iteration converges for every regular partitioning $S_1, \dots, S_p$.*

To prepare our convergence theorem for the overlapping case, we must adapt flow compatibility to decompositions rather than partitionings. The following definition is a natural extension of block flow compatibility for partitionings (cf. Definition 5.6) to decompositions.

DEFINITION 5.11. *Let $A \in \mathbb{R}^{n\times n}$ be an STM-matrix, and let π be a flow compatible numbering of a spanning tree $\mathcal{T} \subset \Gamma(A^T)$. A regular decomposition $S_1, \dots, S_p$ of $(\pi(1), \dots, \pi(n))$ is said to be* ms-compatible *(ms for "multiplicative Schwarz") if there exists a block flow compatible partitioning $\tilde{S}_1, \dots, \tilde{S}_p$ such that $\tilde{S}_i \subseteq S_i$ for all $i = 1, \dots, p$ and if, in addition, $\pi(n) \notin S_j$ for $j = 1, \dots, p-1$.*

The restriction for the root residing in just one set will be discussed after the next theorem. As a preparation, we need the following corollary to Lemma 5.5 for the case that $V \subset \{1, \dots, n\}$ is arbitrary.

COROLLARY 5.12. *Let V be any subset of $\{\pi(1), \dots, \pi(n)\}$ such that $M^{-1} = A[V]^{-1}$ exists.*

(1) *If $\pi(n) \notin V$, then the assertions* (2), (3), *and* (4) *of Lemma* 5.5 *hold.*

(2) *If $\pi(n) \in V$, define $\mathcal{T}_V := \mathcal{T} \cap \Gamma(M^T)$:*

$$V_{\mathcal{T}} := \{\pi(k) \in V : \pi(k) \to_{\mathcal{T}_V} \pi(n)\} \cup \{\pi(n)\},$$
$$V_{\neg\mathcal{T}} := V \setminus V_{\mathcal{T}}.$$

(i) *Then the assertions* (2), (3), *and* (4) *of Lemma* 5.5 *hold for $V_{\neg\mathcal{T}}$, and*

(ii) *the assertions* (1), (3), *and* (5) *of Lemma* 5.5 *hold for $V_{\mathcal{T}}$ with an index $i_0 \in \{1, \dots, n\}$ satisfying $i_0 < n$.*

Proof. Assertion (1) follows as in the proof of Lemma 5.5. As $\pi(n) \notin V_{\neg T}$, there is also nothing to prove for $V_{\neg T}$, i.e., for assertion 2, part (i).

To prove assertion 2, part (ii), consider the guard index $\pi(j_0)$ of $\pi(n) \in V_{\mathcal{T}}$. If $\pi(j_0) \notin V$, then Lemma 5.5 applies directly to $V_{\mathcal{T}}$ with $i_0 = j_0$. Thus assume $\pi(j_0) \in V$. Then an argument analogous to part (2) of the proof of Lemma 5.5 can be used which gives the proposition for some i_0 satisfying $i_0 < n$ and $\pi(i_0) \notin V$. □

THEOREM 5.13. *Let $A \in \mathbb{R}^{n\times n}$ be an STM-matrix, π a flow compatible numbering of a spanning tree $\mathcal{T} \subset \Gamma(A^T)$, and $S_1, \dots, S_p$ an ms-compatible decomposition. Let $(I - P_i)$ be given by* (4.2)*. Then the product*

$$P := (I - P_1) \cdots (I - P_p)$$

has at least one positive column.

Proof. Let $\tilde{S}_1, \ldots, \tilde{S}_p$ be the corresponding partitioning according to Definition 5.11, and set

$$\bar{S}_j := \bigcup_{l=j}^{p} \tilde{S}_l, \quad j = 1, \ldots, p.$$

Put

$$\mathcal{P}_j := (I - P_j) \cdots (I - P_p).$$

By induction we now show that for $j = p, \ldots, 1$

(5.3) there is an l_0 such that for all $\pi(k) \in \bar{S}_j$ there holds $(\pi(k), \pi(l_0)) \in \Gamma(\mathcal{P}_j)$.

Note that $\bar{S}_p = \tilde{S}_p$ and $\tilde{S}_p = \{\pi(k_p), \pi(k_p + 1), \ldots, \pi(n)\}$ for $k_p < n$. Corollary 5.12 implies that there is an index $l_0 < n$ such that $(\pi(j), \pi(l_0)) \in \Gamma(I - P_p) = \Gamma(\mathcal{P}_p)$ for all $j = k_p, \ldots, n$.

Let $p > j > 1$ and assume that (5.3) holds for some j. The inductive step has two parts.

Increasing number of positive elements. Let $\tilde{S}_{j-1} = \{\pi(k_{j-1}), \pi(k_{j-1} + 1), \ldots, \pi(k_{j-1} + l)\}$ for some $l \in \mathbb{N}_0$. Due to Lemma 5.5, there exists for each $i = 0, \ldots, l$ an index $l_i > k_{j-1} + i$ such that $(\pi(k_{j-1} + i), \pi(l_i)) \in \Gamma(I - P_{j-1})$. Since $\pi(n) \notin S_{j-1}$ one gets $l_i > k_{j-1} + l$ for all $i = 0, \ldots, l$. But $k_{j-1} + l = k_j - 1$ and thus $n \geq l_i \geq k_j$ for all $i = 0, \ldots, l$. By the induction hypothesis there is an edge $(\pi(l_i), \pi(l_0)) \in \Gamma(\mathcal{P}_j)$ for all $i = 0, \ldots, l$. Since $\Gamma(\mathcal{P}_{j-1}) = \Gamma(I - P_{j-1})\Gamma(\mathcal{P}_j)$, the relation $(\pi(k_{j-1} + i), \pi(l_0)) \in \Gamma(\mathcal{P}_{j-1})$ follows for all $i = 0, \ldots, l$.

Positivity preservation. Let $\pi(k_0) \in \bar{S}_j = \{\pi(k_j), \ldots, \pi(n)\}$ be arbitrary but fixed. Then by the induction hypothesis $(\pi(k_0), \pi(l_0)) \in \Gamma(\mathcal{P}_j)$. There are two possible cases.

In the case $\pi(k_0) \notin S_{j-1}$, there is an edge $(\pi(k_0), \pi(k_0)) \in \Gamma(I - P_{j-1})$ by the construction of P_{j-1}. But then $(\pi(k_0), \pi(l_0)) \in \Gamma(\mathcal{P}_{j-1})$ as the positive diagonal element in $I - P_{j-1}$ preserves the entry. In the case $\pi(k_0) \in S_{j-1}$, Lemma 5.5 implies that there is an edge $(\pi(k_0), \pi(l_1)) \in \Gamma(I - P_{j-1})$, and $l_1 > k_0$. Consequently, $\pi(l_1) \in \bar{S}_j$. By the induction hypothesis there is an edge $(\pi(l_1), \pi(l_0)) \in \Gamma(\mathcal{P}_j)$, and therefore $(\pi(k_0), \pi(l_0)) \in \Gamma(\mathcal{P}_{j-1})$.

We have thus shown that (5.3) holds for $j = p, \ldots, 1$.

Taking $j = 1$, $\bar{S}_1 = \{\pi(1), \ldots, \pi(n)\}$ and $(\pi(j), \pi(l_0)) \in \Gamma(\mathcal{P}_1)$ for all $j = 1, \ldots, n$. Hence the theorem is proven. ∎

The condition $\pi(n) \notin S_j$ for $j \neq p$ in Definition 5.11 appears to be restrictive. It cannot, however, be omitted when general overlap is allowed. This has been shown by the example from section 4. There are alternative conditions for which we can prove Theorem 5.13 holds. For a given regular decomposition $S_1, \ldots, S_p$, these conditions are as follows.

(1) Every set S_i containing the root can be written as $S_i = \{\pi(k_i), \pi(k_i + 1), \ldots, \pi(n)\}$ for an index $1 \leq k_i \leq n$; i.e., the sets have no gap.

(2) All sets $S_{i_1}, \ldots, S_{i_k}$ containing the root have access to the same index in $\Gamma(A)$; i.e., there exists an index $\pi(i_0) \notin S_{i_j}$ for $j = 1, \ldots, k$, such that $\pi(n) \longrightarrow \pi(i_0)$ in $\Gamma(A)$.

Results based on these conditions are not presented here. Note that neither condition is satisfied in the example from section 4.

5.3. Application to multiplicative Schwarz. Now we have elements in place to prove the convergence of multiplicative Schwarz to a solution of the model problem *MP*; cf. Definition 3.1.

THEOREM 5.14. *Let $A \in \mathbb{R}^{n\times n}$ be an STM-matrix and let $S_1, \dots, S_p$ be an ms-compatible decomposition. Then the one level multiplicative Schwarz iteration* (2.1) *using T_μ given by* (4.3) *converges to a solution of $Ax = b$ for every given $x^0 \in \mathbb{R}^n$ whenever $b \in \mathcal{R}(A)$.*

Proof. The operator T_μ has a positive column by Theorem 5.13. Hence, Corollary 3.7 implies that it is an ST-matrix and the semiconvergence follows from Theorem 5.1. Theorem 5.1 implies also that $\dim \mathcal{N}(I - T_\mu) = 1$. Thus one gets $\mathcal{N}(I - T_\mu) = \mathcal{N}(A)$ and the iteration is consistent. ∎

6. Damped projections. Now we investigate multiplicative Schwarz iterations for damped projections; i.e., for an STM-matrix A we consider the operator

$$T_{\mu,\eta} := (I - \eta P_1) \cdots (I - \eta P_p) \tag{6.1}$$

for a regular decomposition $S_1, \dots, S_p$ and some $\eta \in (0, 1)$. We choose $\eta \in (0, 1)$ since we can view this as introducing under relaxation because

$$I - \eta P_i = (1 - \eta)I + \eta(I - P_i)$$

for all $i = 1, \dots p$.

Proposition 4.1 applies verbatim to $T_{\mu,\eta}$. Trivially, since $\eta \in (0, 1)$, it follows that $T_{\mu,\eta}$ has a positive diagonal; thus it is semiconvergent. Hence, it remains only to prove consistency.

We will show that $T_{\mu,\eta}$ is an ST-matrix. Then we are done, because the ST property implies consistency. We need the following lemma, which transforms the question of consistency from a multiplicative problem into an additive one and follows in a straightforward manner from Lemma 2.1.

LEMMA 6.1. *Consider nonnegative square matrices $L_1, \dots, L_p$ and assume that the diagonal of each L_i is positive. Then for any permutation $\sigma : \{1, \dots, p\} \longrightarrow \{1, \dots, p\}$,*

$$\Gamma\left(\sum_{j=1}^{p} L_{\sigma(j)}\right) = \bigcup_{j=1}^{p} \Gamma(L_{\sigma(j)}) \subset \Gamma(L_{\sigma(1)} \cdots L_{\sigma(p)}).$$

We can now prove the desired result.

THEOREM 6.2. *Suppose $A \in \mathbb{R}^{n\times n}$ is an STM-matrix, α is the final and basic class of A, and $i_0 \in \alpha$ is arbitrary. Let $S_1, \dots, S_p$ be a regular decomposition such that i_0 is contained in exactly one set S_j. Then*

$$T^{(\sigma)}_{\mu,\eta} := (I - \eta P_{\sigma(1)}) \cdots (I - \eta P_{\sigma(p)})$$

is a semiconvergent ST-matrix for any permutation $\sigma : \{1, \dots, p\} \longrightarrow \{1, \dots, p\}$ and $\eta \in (0, 1)$.

Proof. By Lemma 6.1 and the nonnegativity of the damped projections, it suffices to show that the "flow" given by $\mathcal{T}$ is preserved, i.e., that the graph of

$$\sum_{j=1}^{p} (I - \eta P_j)$$

contains a spanning tree with a guard.

To prove this let $\mathcal{T} \subset \Gamma(A^T)$ be an arbitrary spanning tree with root i_0 (cf. Corollary 3.4). Consider a flow compatible numbering π of $\mathcal{T}$; then $i_0 = \pi(n)$. Assume w.l.o.g. that $\pi(n) \in S_p$ and define

$$\tilde{S}_p := \{\pi(j) \in S_p : \pi(j) \to_{\mathcal{T}} \pi(n)\} \cup \{\pi(n)\}. \tag{6.2}$$

By Corollary 5.12 there exists an index j_0 such that $j_0 < n$ and each $\pi(j) \in \tilde{S}_p$ has access to $\pi(j_0) \notin \tilde{S}_p$ in $\Gamma(I - \eta P_p)$. Thus, it remains to show that each $\pi(l) \in \{1, \ldots, n\} \setminus \tilde{S}_p$ has access to some $\pi(j) \in \tilde{S}_p$ in

$$\Gamma\left(\sum_{j=1}^{p}(I - \eta P_j)\right). \tag{6.3}$$

Therefore, we let $\pi(l_0) \in \{1, \ldots, n\} \setminus \tilde{S}_p$ be arbitrary but fixed. Then $\pi(l_0) \in S_{k_0} \setminus \tilde{S}_p$ for some $k_0 \in \{1, \ldots, p\}$. It follows from Lemma 5.5 that $\pi(l_0)$ has access to some $\pi(l_1)$ in $\Gamma(I - \eta P_{k_0})$ and $l_1 > l_0$. If $\pi(l_1) \notin \tilde{S}_p$, then $\pi(l_1)$ has access to some $\pi(l_2)$ in $\Gamma(I - \eta P_{k_1})$ for some $k_1 \in \{1, \ldots, p\}$ and $l_2 > l_1$. So, inductively, after a maximum of $n-1$ steps, it follows that $\pi(l_0)$ has access to some $\pi(j) \in \tilde{S}_p$ in the graph from (6.3).

Consequently, each $\pi(j) \in \{1, \ldots, n\}$ has access to $\pi(j_0)$ and since $\pi(j_0) \in \{1, \ldots, n\} \setminus \tilde{S}_p$, $\pi(j_0)$ must have access to itself; i.e. the graph (6.3) contains a spanning tree with a guard index. Lemma 6.1 implies that this also holds for $T_{\mu,\eta}^{(\sigma)}$. As $T_{\mu,\eta}^{(\sigma)} z = z$ for the positive vector $z \in \mathcal{N}(A)$, we have from Corollary 3.7 that $T_{\mu,\eta}^{(\sigma)}$ is an ST-matrix. Additionally, the diagonals of $T_{\mu,\eta}^{(\sigma)}$ are positive; thus $T_{\mu,\eta}^{(\sigma)}$ is semiconvergent. □

Remark 6.1. We do not know whether one could dispense with the condition that one index from α is contained in just one set S_j. In the case of an irreducible matrix, this condition reduces to having one variable which has no overlap, which will usually be fulfilled.

DEFINITION 6.3. *Let $A = I - B \in \mathbb{R}^{n\times n}$ be an STM-matrix and denote by α its basic class. A regular decomposition $S_1, \ldots, S_p$ with respect to A is said to be* root preserving *if there exists an $i_0 \in \alpha$ such that $|\{j : i_0 \in S_j\}| = 1$.*

The following theorem is a simple application of Theorem 6.2 and gives us the convergence of the damped Schwarz iteration for *MP*.

THEOREM 6.4. *Let $A \in \mathbb{R}^{n\times n}$ be an STM-matrix, and let $S_1, \ldots, S_p$ be a root preserving decomposition. Then the damped one level multiplicative Schwarz iteration, i.e., iteration* (2.1) *using $T_{\mu,\eta}$ given by* (6.1)*, converges to the solution of $Ax = b$ for every given $x^0 \in \mathbb{R}^n$ whenever $b \in \mathcal{R}(A)$ and $\eta \in (0, 1)$.*

7. Convergence of additive Schwarz for *MP*. Finally, we investigate the convergence of additive Schwarz iterations for the model problem *MP* given by the iteration (2.1) and (4.4). This is now quite simple using the theory already developed.

Let us consider a root preserving decomposition $S_1, \ldots, S_p$ from the last section. We know that

$$\sum_{j=1}^{p}(I - \eta P_j) = pI - \eta P_1 - \cdots - \eta P_p \tag{7.1}$$

is a semiconvergent ST-matrix by the proof of Theorem 6.2. Comparing (7.1) and (4.4) reveals that T_θ is also an ST-matrix because

$$\Gamma(T_\theta) = \Gamma\left(\sum_{j=1}^{p}(I - \eta P_j)\right).$$

The semiconvergence of T_θ has been considered in Proposition 4.1. Thus we have the following theorem (which obviously applies to MP).

THEOREM 7.1. *Let $A \in \mathbb{R}^{n\times n}$ be an STM-matrix and let $S_1, \ldots, S_p$ be a root preserving decomposition. Then the one level additive Schwarz iteration, i.e., iteration* (2.1) *using T_θ given by* (4.4)*, converges to the solution of $Ax = b$ for every given $x^0 \in \mathbb{R}^n$ whenever $b \in \mathcal{R}(A)$ and $\theta \in (0, 1/q)$, where q is the measure of overlap.*

8. Concluding remarks and outlook. In this paper we introduce a graph theoretical approach and prove new convergence theorems for the classical one level multiplicative Schwarz iteration, the damped one level multiplicative Schwarz iteration, and the one level additive Schwarz method when applied to singular M-matrices.

The major ingredient is a compatibility condition between a spanning tree embedded in the graph of the matrix and the sets defining the Schwarz decomposition. In the case of overlap, additional conditions had to be imposed on the root of the tree. It is not possible to completely ignore these conditions, as we have shown in a counterexample.

Several generalizations are possible [5]. We can weaken condition (1.2) to

$$Az = 0 \quad \text{for some positive vector } z;$$

i.e., we allow the dimension of the null space of A to be larger than one. Indeed, the results presented here carry over with only minor restrictions to this new situation. Additionally, within our framework we can also obtain convergence results of Schwarz methods for two-stage (see, e.g., [6, 17, 21]) and partially asynchronous (see, e.g., [2, 7, 14, 12]) variants. These results are presented in a future companion paper.

Finally we mention that the graph based approach used here is quite different from those in [10, 18, 23, 25]. In those references, only splittings of a singular M-matrix without overlap are discussed.

REFERENCES

[1] G. ALEFELD AND H. SCHNEIDER, *On square roots of M-matrices*, Linear Algebra Appl., 42 (1982), pp. 73–132.

[2] J. BAHI, *Asynchronous iterative algorithms for nonexpansive linear systems*, J. Parallel and Distributed Computing, 60 (2000), pp. 92–112.

[3] M. BENZI, A. FROMMER, R. NABBEN, AND D. B. SZYLD, *Algebraic theory of multiplicative Schwarz methods*, Numer. Math., 89 (2001), pp. 605–639.

[4] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Science*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.

[5] S. BOROVAC, *A Graph Based Approach to the Convergence of Some Iterative Methods for Singular M-Matrices and Markov Chains*, Ph.D. dissertation, Universität Wuppertal, Wuppertal, Germany, 2006. Available online at elpub.bib.uni-wuppertal.de/edocs/dokumente/fbc/mathematik/diss2006/borovac.

[6] R. BRU, L. ELSNER, AND M. NEUMANN, *Convergence of infinite products of matrices and inner outer iteration schemes*, Electron. Trans. Numer. Anal., 2 (1995), pp. 183–193.

[7] R. BRU, V. MIGALLÓN, J. PENADÉS, AND D. B. SZYLD, *Parallel synchronous and asynchronous two-stage multisplitting methods*, Electron. Trans. Numer. Anal., 3 (1995), pp. 24–38.

[8] R. Bru, F. Pedroche, and D. B. Szyld, *Additive Schwarz iterations for Markov chains*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 445–458.
[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed., MIT Press, Cambridge, MA, McGraw–Hill, Boston, 2001.
[10] P. J. Courtois and P. Semal, *Block iterative algorithms for stochastic matrices*, Linear Algebra Appl., 76 (1986), pp. 59–70.
[11] R. Diestel, *Graph Theory*, 3rd ed., Springer-Verlag, Berlin, Heidelberg, New York, 2005.
[12] A. Frommer and D. B. Szyld, *Asynchronous two-stage iterative methods*, Numer. Math., 69 (1994), pp. 141–153.
[13] A. Frommer and D. B. Szyld, *Weighted max norms, splittings, and overlapping additive Schwarz iterations*, Numer. Math., 83 (1999), pp. 259–278.
[14] A. Frommer and D. B. Szyld, *On asynchronous iterations*, J. Comput. Appl. Math., 23 (2000), pp. 201–216.
[15] J. Hajnal, *Weak ergodicity in non-homogeneous Markov chains*, Proc. Cambridge Philos. Soc., 54 (1958), pp. 233–246.
[16] I. Marek and D. B. Szyld, *Algebraic Schwarz methods for the numerical solution of Markov chains*, Linear Algebra Appl., 386 (2004), pp. 67–81.
[17] V. Migallón, J. Penadés, and D. B. Szyld, *Block two-stage methods for singular systems and Markov chains*, Linear Algebra Appl., 3 (1996), pp. 413–426.
[18] D. Mitra and P. Tsoucas, *Relaxations for the numerical solution of some stochastic problems*, Comm. Statist. Stochastic Models, 4 (1988), pp. 387–419.
[19] R. Nabben, *Comparisons between additive and multiplicative Schwarz iterations in domain decomposition methods*, Numer. Math., 93 (2003), pp. 145–162.
[20] R. Nabben and D. B. Szyld, *Schwarz iterations for symmetric positive semidefinite problems*, SIAM J. Matrix Anal. Appl., 29 (2006), pp. 98–116.
[21] N. K. Nichols, *On the convergence of two-stage iterative processes for solving linear equations*, SIAM J. Numer. Anal., 10 (1973), pp. 460–469.
[22] A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, Clarendon Press, Oxford, UK, 1999.
[23] D. J. Rose, *Convergent regular splitting for singular M-matrices*, SIAM J. Alg. Disc. Meth., 5 (1984), pp. 133–144.
[24] U. G. Rothblum, *Algebraic eigenspaces of nonnegative matrices*, Linear Algebra Appl., 12 (1975), pp. 281–292.
[25] H. Schneider, *Theorems on M-splittings of a singular M-matrix which depend on graph structure*, Linear Algebra Appl., 58 (1984), pp. 407–424.
[26] E. Seneta, *Non-negative Matrices and Markov Chains*, 2nd ed., Springer-Verlag, New York, Berlin, Heidelberg, 1981.
[27] B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1996.
[28] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.
[29] R. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput., 1 (1972), pp. 146–160.
[30] A. Toselli and O. B. Widlund, *Domain Decomposition: Algorithms and Theory*, Springer-Verlag, New York, Berlin, Heidelberg, 2004.
[31] R. S. Varga, *Matrix Iterative Analysis. Second Revised and Expanded Edition*, Springer-Verlag, New York, Berlin, Heidelberg, 2000.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1392–1405

NEW CRITERIA FOR THE FEASIBILITY OF THE CHOLESKY METHOD WITH INTERVAL DATA*

G. ALEFELD† AND G. MAYER‡

Dedicated to Prof. Dr. Ulrich Kulisch of the University of Karlsruhe on the occasion of his 75th birthday

Abstract. We present some new criteria for the feasibility of the interval Cholesky method. In particular, we relate this feasibility to that of the interval Gaussian algorithm.

Key words. linear interval equations, Cholesky method, interval Cholesky algorithm, Gaussian algorithm, interval Gaussian algorithm, linear system of equations, criteria of feasibility

AMS subject classifications. 65G20, 65G30

DOI. 10.1137/070711232

1. Introduction.

In [2] we introduced the interval Cholesky method in order to enclose the symmetric solution set

$$S_{\text{sym}} = \{x \in \mathbb{R}^n |\ Ax = b,\ A = A^T \in [A] = [A]^T,\ b \in [b]\},$$

where $[A] = [\underline{A}, \overline{A}]$ is a given $n \times n$ interval matrix and $[b]$ is a corresponding interval vector. The algorithm uses the formulae of the classical Cholesky method, replacing the real entries and arithmetic by interval ones. It terminates with an interval vector $[x]^C = \text{ICh}([A], [b])$ which encloses S_{sym} but not necessarily the general solution set

$$S = \{x \in \mathbb{R}^n |\ Ax = b,\ A \in [A],\ b \in [b]\},$$

which also contains the solutions of linear systems with unsymmetric matrices from $[A]$. A criterion necessary for $[x]^C$ to exist is the positive definiteness of all symmetric matrices in $[A]$—independently of any right-hand side $[b]$. Unfortunately, this property is not sufficient as Reichmann's example in [13] shows which originally was constructed for a different situation. This example caused the necessity of criteria which guarantee the existence of $[x]^C$ or, equivalently, the feasibility of the interval Cholesky method for arbitrary right-hand interval sides. In [2] we proved that $[x]^C$ exists for a variety of structured matrices, among them H-matrices, M-matrices, diagonal dominant matrices, and tridiagonal ones, all with appropriate additional properties. In [3] we extended these criteria of feasibility by perturbation results analogously to those in [11]. In [15] further results of feasibility were presented for block variants of the algorithm which were introduced there. It is the purpose of the present paper to add others. In particular, we will show that the feasibility of the interval Gaussian algorithm [1] implies the existence of $[x]^C$ provided that $[A]$ contains at least one positive definite element matrix. Based on this crucial result a

*Received by the editors December 19, 2007; accepted for publication (in revised form) by A. Frommer June 19, 2008; published electronically October 22, 2008.
http://www.siam.org/journals/simax/30-4/71123.html

†Institut für Angewandte und Numerische Mathematik, Universität Karlsruhe (Karlsruhe Institute of Technology – KIT), Kaiserstr. 12, D–76128 Karlsruhe, Germany (goetz.alefeld@mathematik.uni-karlsruhe.de).

‡Institut für Mathematik, Universität Rostock, Universitätsplatz 1, D–18051 Rostock, Germany (guenter.mayer@uni-rostock.de).

great deal of criteria for the interval Gaussian algorithm carries over to the interval Cholesky method. Unfortunately, the feasibility of the interval Cholesky method does not necessarily imply that of the interval Gaussian algorithm. We will illustrate this phenomenon by an example. It was unexpected, since we can show that the existence of $[x]^C$ for each *symmetric* matrix $\tilde{A} \in [A]$ implies the feasibility of the Gaussian algorithm for *any* matrix $A \in [A]$ and not only for the symmetric ones.

We have organized this paper as follows: In section 2 we recall the formulae for the algorithm and a recursive representation. In addition we introduce our notation and some basic facts that are used later on. In section 3 we state and prove our new results illustrating them by examples.

2. Preliminaries. By $\mathbb{R}^n, \mathbb{R}^{n\times n}, \mathbb{IR}, \mathbb{IR}^n, \mathbb{IR}^{n\times n}$ we denote the set of real vectors with n components, the set of real $n\times n$ matrices, the set of intervals, the set of interval vectors with n components, and the set of $n \times n$ interval matrices, respectively. By "interval" we always mean a real compact interval. We write interval quantities in brackets with the exception of point quantities (i.e., degenerate interval quantities) which we identify with the element they contain. Examples are the zero matrix O, the identity matrix I, and the vector $e = (1, 1, \ldots, 1)^T$. We use the notation $[A] = [\underline{A}, \overline{A}] = ([a]_{ij}) = ([\underline{a}_{ij}, \overline{a}_{ij}]) \in \mathbb{IR}^{n\times n}$ simultaneously without further reference, and we proceed similarly for the elements of $\mathbb{R}^n, \mathbb{R}^{n\times n}, \mathbb{IR}$, and $\mathbb{IR}^n$. We also mention the standard notation from interval analysis ([1], [11]),

$$\check{a} = \operatorname{mid}([a]) = (\underline{a} + \overline{a})/2 \qquad \text{(midpoint)},$$

$$|[a]| = \max\{|\tilde{a}| \mid \tilde{a} \in [a]\} = \max\{|\underline{a}|, |\overline{a}|\} \qquad \text{(absolute value)},$$

$$\langle [a] \rangle = \min\{|\tilde{a}| \mid \tilde{a} \in [a]\} = \begin{cases} \min\{|\underline{a}|, |\overline{a}|\} & \text{if } 0 \notin [a], \\ 0 & \text{otherwise} \end{cases} \qquad \text{(minimal absolute value)}$$

for intervals $[a]$. For $[A] \in \mathbb{IR}^{n\times n}$ we obtain $|[A]| \in \mathbb{R}^{n\times n}$ by applying the operator $|\cdot|$ entrywise, and we define the comparison matrix $\langle [A] \rangle = (c_{ij}) \in \mathbb{R}^{n\times n}$ by setting

$$c_{ij} = \begin{cases} -|[a]_{ij}| & \text{if } i \neq j, \\ \langle [a]_{ii} \rangle & \text{if } i = j. \end{cases}$$

Since real numbers can be viewed as degenerate intervals, $|\cdot|$ and $\langle\cdot\rangle$ can also be used for them. In this case they coincide with their well known real counterpart.

By $A \geq O$ we denote a nonnegative $n \times n$ matrix, i.e., $a_{ij} \geq 0$ for $i, j = 1, \ldots, n$. Analogously, we define $x \geq 0$ for $x \in \mathbb{R}^n$. We call $x \in \mathbb{R}^n$ positive writing $x > 0$ if $x_i > 0$, $i = 1, \ldots, n$. We use $Z^{n\times n}$ for the set of real $n \times n$ matrices with nonpositive off-diagonal entries. Trivially, $Z^{n\times n}$ contains the $n \times n$ matrix $\langle A \rangle$. As usual we call $A \in \mathbb{R}^{n\times n}$ an M-matrix if A is nonsingular with $A^{-1} \geq O$ and $A \in Z^{n\times n}$. It is an H-matrix if $\langle A \rangle$ is an M-matrix.

An interval matrix $[A] \in \mathbb{IR}^{n\times n}$ is defined to be an M-matrix if each element $\tilde{A} \in [A]$ is an M-matrix. In the same way the term "H-matrix" can be extended to $\mathbb{IR}^{n\times n}$. It is easy to verify that $[A] \in \mathbb{IR}^{n\times n}$ is an M-matrix if and only if $\underline{A}$ is an M-matrix and $\overline{a}_{ij} \leq 0$ for $i \neq j$, and that $[A] \in \mathbb{IR}^{n\times n}$ is an H-matrix if and only if $\langle [A] \rangle$ is an M-matrix.

We call $[A] \in \mathbb{IR}^{n\times n}$ irreducible if $\langle [A] \rangle$ is irreducible. In the same way we define $[A]$ to be diagonally dominant, strictly diagonally dominant, and irreducibly

diagonally dominant, respectively. If there is a positive vector x such that

$$\langle [A] \rangle x \geq 0 \tag{2.1}$$

holds, then we call $[A]$ generalized diagonally dominant. Moreover, we define $[A]$ to be generalized strictly diagonally dominant if strict inequality holds in (2.1). Analogously, a generalized irreducibly diagonally dominant matrix $[A]$ is irreducible and generalized diagonally dominant with $(\langle [A] \rangle x)_i > 0$ in (2.1) for at least one component i. It is well known that generalized strictly diagonally dominant matrices are H-matrices and vice versa.

We equip $\mathbb{IR}, \mathbb{IR}^n, \mathbb{IR}^{n \times n}$ with the usual real interval arithmetic as described in [1], [11]. We assume that the reader is familiar with the basic properties of this arithmetic. For $[a] \in \mathbb{IR}$ we define

$$\sqrt{[a]} = \{\sqrt{a} \mid a \in [a]\} \text{ for } 0 \leq \underline{a}$$

and

$$[a]^2 = \{a^2 \mid a \in [a]\}. \tag{2.2}$$

Instead of $\sqrt{[a]}$ we also write $[a]^{1/2}$.

Then the interval Cholesky method reads as follows.

Given $[A] = [A]^T \in \mathbb{IR}^{n \times n}$ and $[b] \in \mathbb{IR}^n$, define the lower triangular matrix $[L]$ and the vectors $[y], [x]^C = ([x]_i^C) = \mathrm{ICh}([A], [b]) \in \mathbb{IR}^n$ by

$$[l]_{jj} = \left([a]_{jj} - \sum_{k=1}^{j-1} [l]_{jk}^2 \right)^{1/2},$$

$$[l]_{ij} = \left([a]_{ij} - \sum_{k=1}^{j-1} [l]_{ik}[l]_{jk} \right) / [l]_{jj}, \quad i = j+1, \ldots, n,$$

$$j = 1, \ldots, n,$$

$$[y]_i = \left([b]_i - \sum_{j=1}^{i-1} [l]_{ij}[y]_j \right) / [l]_{ii}, \qquad i = 1, \ldots, n,$$

$$[x]_i^C = \left([y]_i - \sum_{j=i+1}^{n} [l]_{ji}[x]_j^C \right) / [l]_{ii}, \quad i = n, n-1, \ldots, 1.$$

Sums with an upper bound smaller than the lower one are defined to be zero; the squares in the first formula are evaluated by applying the interval square function (2.2).

Apparently $[x]^C$ exists if and only if $0 < \underline{l}_{ii}$, $i = 1, \ldots, n$. In this case we call the algorithm feasible. Note that this feasibility does not depend on the choice of $[b]$. For the interval Cholesky method we assume, without loss of generality, $[A]$ to be symmetric, i.e., $[A] = [A]^T$. (In the case $[A] \neq [A]^T$ we replace $[A]$ by the largest interval matrix $[B] \subseteq [A]$ which satisfies $[B] = [B]^T$ and rename $[B]$ to $[A]$.) By the overestimation of the interval arithmetic only

$$[A] \subseteq [L][L]^T$$

can be guaranteed; cf. [2] for details. Nevertheless the pair $([L],[L]^T)$ is called the Cholesky decomposition of $[A]$. This decomposition can also be defined in a recursive way. To this end write $[A] \in \mathbb{IR}^{n\times n}$ as

$$[A] = \begin{pmatrix} [a]_{11} & [c]^T \\ [c] & [A]' \end{pmatrix}$$

and use its Schur complement $\Sigma^C_{[A]} = [A]' - [c][c]^T/[a]_{11}$ if $n > 1$, $0 \notin [a]_{11}$, where $[c]_i[c]_i$ is evaluated as $[c]_i^2$.

DEFINITION 2.1 (equivalent definition of $([L],[L]^T)$). *The pair $([L],[L]^T)$ is called the Cholesky decomposition of $[A] = [A]^T \in \mathbb{IR}^{n\times n}$ if $0 < \underline{a}_{11}$ and if either $n = 1$, $[L] = (\sqrt{[a]_{11}})$, or if $n > 1$ and*

$$[L] = \begin{pmatrix} \sqrt{[a]_{11}} & 0 \\ [c]/\sqrt{[a]_{11}} & [L]' \end{pmatrix}, \tag{2.3}$$

where $([L]',([L]')^T)$ is the Cholesky decomposition of $\Sigma^C_{[A]}$. If $0 \in [a]_{11}$, then the Cholesky decomposition does not exist.

In [2] we showed that the matrix $[L]$ in Definition 2.1 is the same as that defined by the interval Cholesky method. In particular, the existence of the Cholesky decomposition is equivalent to the feasibility of the interval Cholesky method. We will exploit this fact later. It is a basic fact of matrix analysis that the existence of the Cholesky decomposition of a symmetric point matrix $A \in \mathbb{R}^{n\times n}$ is equivalent to A being positive definite, to A having only positive eigenvalues, and to A having only positive leading principal minors; cf., for instance, [7].

Directly from the formulae of the interval Cholesky method we obtain the following result which corresponds to Lemma 3.1 (b) in [8].

LEMMA 2.1. *Let $[A] = [A]^T \in \mathbb{IR}^{n\times n}$, $[b] \in \mathbb{IR}^n$, and let $[x]^C = \mathrm{ICh}([A],[b])$ exist. If $D = \mathrm{diag}(d_1,\ldots,d_n) \in \mathbb{R}^{n\times n}$ has positive entries d_i, $i = 1,\ldots,n$, in the diagonal, then $[\tilde{x}]^C = \mathrm{ICh}([D[A]D, D[b])$ exists and satisfies $[\tilde{x}]^C = D^{-1}[x]^C$.*

Proof. Denote by a tilde all items which belong to $[\tilde{x}]^C$. Then, by induction, the formulae of the interval Cholesky method yield $[\tilde{L}] = D[L]$, hence $[\tilde{y}] = [y]$ and $[\tilde{x}]^C = D^{-1}[x]^C$. ∎

We continue by recalling some results from [2].

THEOREM 2.1. *Let $[A] = [A]^T \in \mathbb{IR}^{n\times n}$ be an H-matrix with $0 < \underline{a}_{ii}$, $i = 1,\ldots,n$. Then the following statements hold.*

(a) *The vector $[x]^C$ exists, and $[L]$ is again an H-matrix.*

(b) *Each symmetric matrix $\tilde{A} \in [A]$ is positive definite.*

From Theorem 2.1 we easily get the following corollary.

COROLLARY 2.1. *Let $[A] = [A]^T \in \mathbb{IR}^{n\times n}$ be an H-matrix. Then the following statements are equivalent.*

(i) *The vector $[x]^C$ exists.*

(ii) *The sign condition $\underline{a}_{ii} > 0$, $i = 1,\ldots,n$, holds.*

(iii) *The matrix $[A]$ contains at least one symmetric and positive definite element $\tilde{A} \in [A]$.*

Proof. (i) ⇒ (ii). Since $\langle[A]\rangle$ is an M-matrix we have $\langle[a]_{ii}\rangle > 0$, $i = 1,\ldots,n$, whence $0 \notin [a]_{ii}$. The existence of $[x]^C$ then implies $\underline{a}_{ii} > 0$.

The implications (ii) ⇒ (i) and (ii) ⇒ (iii) follow directly from Theorem 2.1.

(iii) ⇒ (ii). As in the first implication above, one gets $0 \notin [a]_{ii}$, and the sign condition for $\underline{a}_{ii}$ follows from the positive definiteness of $\tilde{A}$. ∎

THEOREM 2.2. *Let $[A] = [A]^T \in \mathbb{IR}^{n\times n}$ be a tridiagonal matrix, and let $\tilde{A} \in [A]$ be any symmetric matrix which satisfies $\langle \tilde{A}\rangle = \langle [A]\rangle$ and is positive definite. Then $[A]$ is an H-matrix; in particular, all symmetric matrices $A \in [A]$ are positive definite, and $[x]^C$ exists.*

THEOREM 2.3. *Let*

$$A = \begin{pmatrix} a_{11} & c^T \\ c & A' \end{pmatrix} \in \mathbb{R}^{n\times n}$$

be symmetric and positive definite. Then the Schur complement $\Sigma_A^C = A' - cc^T/a_{11}$ of A is symmetric and positive definite.

Proof. Use $0 < x^T Ax = (x')^T \Sigma_A^C x'$ for $x = \left(-c^T x'/a_{11},\, (x')^T\right)^T$ and any nonzero vector $x' \in \mathbb{R}^{n-1}$. □

Since we will also use results of the interval Gaussian algorithm we will recall its formulae, too.

Given $[A] \in \mathbb{IR}^{n\times n}$ and $[b] \in \mathbb{IR}^n$, define $[A]^{(k)} = ([a]_{ij}^{(k)}) \in \mathbb{IR}^{n\times n}$, $[b]^{(k)} = ([b]_i^{(k)}) \in \mathbb{IR}^n$, $k = 1,\dots,n$, and $[x]^G = ([x]_i^G) = \mathrm{IGA}([A],[b]) \in \mathbb{IR}^n$ by

$$[A]^{(1)} = [A], \quad [b]^{(1)} = [b],$$

$$[a]_{ij}^{(k+1)} = \begin{cases} [a]_{ij}^{(k)}, & i = 1,\dots,k,\ j = 1,\dots,n, \\ [a]_{ij}^{(k)} - \dfrac{[a]_{ik}^{(k)} \cdot [a]_{kj}^{(k)}}{[a]_{kk}^{(k)}}, & i = k+1,\dots,n,\ j = k+1,\dots,n, \\ 0 & \text{otherwise}, \end{cases}$$

$$[b]_i^{(k+1)} = \begin{cases} [b]_i^{(k)}, & i = 1,\dots,k, \\ [b]_i^{(k)} - \dfrac{[a]_{ik}^{(k)}}{[a]_{kk}^{(k)}} \cdot [b]_k^{(k)}, & i = k+1,\dots,n, \end{cases}$$

$$k = 1,\dots,n-1,$$

$$[x]_i^G = \left([b]_i^{(n)} - \sum_{j=i+1}^{n} [a]_{ij}^{(n)} [x]_j^G\right) \Big/ [a]_{ii}^{(n)}, \quad i = n, n-1,\dots,1.$$

For $i = n$ the sum is set equal to zero.

Note that $[x]^G$ is defined without permuting rows or columns. The algorithm is feasible if and only if $0 \notin [a]_{kk}^{(k)}$, $k = 1,\dots,n$, where again the feasibility does not depend on the choice of $[b]$. Define the lower triangular matrix $[\hat{L}]$ by $[\hat{l}]_{ii} = 1$, $[\hat{l}]_{ij} = [a]_{ij}^{(j)}/[a]_{jj}^{(j)}$ for $i > j$, and the upper triangular matrix $[\hat{U}]$ by $[\hat{u}]_{ij} = [a]_{ij}^{(n)}$ for $i \le j$. According to [11] the pair $([\hat{L}],[\hat{U}])$ is called the triangular decomposition of $[A]$.

Similar to Definition 2.1 there is an equivalent recursive definition of that decomposition. It uses the partition

$$[A] = \begin{pmatrix} [a]_{11} & [c]^T \\ [d] & [A]' \end{pmatrix}$$

and its Schur complement $\Sigma^G_{[A]} = [A]' - [d][c]^T/[a]_{11}$ if $n > 1$, $0 \notin [a]_{11}$. Note that for $[A] = [A]^T$ we have $[c] = [d]$. In this case we assume that $[c]_i[c]_i$ in the product $[d][c]^T = [c][c]^T$ is evaluated as a product of intervals and not as in (2.2). This implies

$$\Sigma^C_{[A]} \subseteq \Sigma^G_{[A]}, \tag{2.4}$$

where both matrices may differ from each other. For symmetric *point matrices* $A \equiv [A]$, however, equality always holds in (2.4), provided that $a_{11} > 0$.

DEFINITION 2.2 (equivalent definition of $([\hat{L}], [\hat{U}])$). *The pair* $([\hat{L}], [\hat{U}])$ *is called triangular decomposition of* $[A] \in \mathbb{IR}^{n\times n}$ *if* $0 \notin [a]_{11}$ *and if either* $n = 1$, $[\hat{L}] = 1$, $[\hat{U}] = ([a]_{11})$, *or if* $n > 1$ *and*

$$[\hat{L}] = \begin{pmatrix} 1 & 0 \\ [d]/[a]_{11} & [\hat{L}]' \end{pmatrix}, \qquad [\hat{U}] = \begin{pmatrix} [a]_{11} & [c]^T \\ 0 & [\hat{U}]' \end{pmatrix},$$

where $([\hat{L}]', [\hat{U}]')$ *is the triangular decomposition of* $\Sigma^G_{[A]}$. *If* $0 \in [a]_{11}$, *then the triangular decomposition does not exist.*

In what follows we will use the notation of section 2 without further reference.

3. New results. In this section we will present some new criteria for the feasibility of the interval Cholesky method. Since neither the existence of $[x]^C$ nor that of $[x]^G$ depends on the right-hand side $[b]$, we do not refer to $[b]$ in our results.

Assume now that $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite. Then from the Cholesky decomposition (L, L^T) of A we define the diagonal matrix $D = \mathrm{diag}(l_{11}, \ldots, l_{nn})$. It is well known that D has positive diagonal entries. Hence $A = LL^T = (LD^{-1})(DL^T)$ yields the unique $(\hat{L}, \hat{U})$-decomposition of A with $\hat{L} = LD^{-1}$ and $\hat{U} = DL^T$. Conversely, from the $(\hat{L}, \hat{U})$-decomposition of a symmetric matrix $A \in \mathbb{R}^{n\times n}$ with positive diagonal entries $\hat{u}_{ii}$, $i = 1, \ldots, n$, one easily verifies positive definiteness of A and hence the existence of the Cholesky decomposition. Therefore, the question arises at once whether a similar result also holds in the interval case. In one direction the answer is positive.

THEOREM 3.1. *Let* $[A] = [A]^T \in \mathbb{IR}^{n\times n}$ *contain a symmetric and positive definite matrix* $\tilde{A}$. *If* $[x]^G$ *exists, then* $[x]^C$ *exists, too.*

Proof. Since $\tilde{A} \in [A]$ is symmetric and positive definite we have $\tilde{a}_{11} > 0$. Moreover, since by assumption $[x]^G$ exists we obtain $\underline{a}_{11} > 0$. We now proceed by induction on the dimension of $[A]$.

If $n = 1$, then the assertion is obvious. If $n > 1$, then let it hold for dimensions less than n, and let

$$[A] = \begin{pmatrix} [a]_{11} & [c]^T \\ [c] & [A'] \end{pmatrix}.$$

From $\underline{a}_{11} > 0$, the Schur complements

$$\Sigma^C_{[A]} = [A]' - [c][c]^T/[a]_{11} \quad (\text{with } [c]_i[c]_i \text{ being evaluated as } [c]_i^2)$$

for the Cholesky method and

$$\Sigma^G_{[A]} = [A]' - [c][c]^T/[a]_{11} \supseteq \Sigma^C_{[A]}$$

(with $[c]_i[c]_i$ being evaluated as a product of two intervals) for the Gaussian algorithm exist. Since, by assumption, the interval Gaussian algorithm is feasible for $\Sigma^G_{[A]}$

and since this interval matrix contains the symmetric and positive definite matrix $\Sigma^G_{\tilde{A}} = \Sigma^C_{\tilde{A}}$ (cf. Theorem 2.3), the induction hypothesis applies for $\Sigma^G_{[A]}$. Therefore, the Cholesky decomposition exists for this interval matrix and thus exists for the (possibly proper) subset $\Sigma^C_{[A]}$, too. □

We will prove now a result on point matrices which originally increased our hope for a converse of Theorem 3.1.

THEOREM 3.2. *Let all symmetric matrices $\tilde{A} \in [A] = [A]^T \in \mathbb{IR}^{n\times n}$ be positive definite. Then the Gaussian algorithm is feasible without pivoting for all matrices $A \in [A]$ (and not only for the symmetric ones).*

Proof. Let $A \in [A]$. Then the symmetric part[1] $A_{\mathrm{sym}} = (A + A^T)/2$ of A is contained in $[A]$, and hence it is positive definite by assumption. For $x \neq 0$ we have

$$0 < x^T A_{\mathrm{sym}} x = (x^T Ax + x^T A^T x)/2 = x^T Ax, \tag{3.1}$$

where we used $x^T A^T x = (x^T A^T x)^T = x^T Ax$. From (3.1) we immediately get $\det A \neq 0$. Since this implication applies also to all leading submatrices of A the assertion follows from Theorem 9.1.2 in [12]. □

Despite this positive result the converse of Theorem 3.1 does not hold. This is illustrated by the following example.

Example 3.1. Consider the matrix

$$[A] = \begin{pmatrix} 1 & [-1,1] & 0 & 0 \\ [-1,1] & 2 & 1 & 2 \\ 0 & 1 & 2 & 2 \\ 0 & 2 & 2 & 5+\varepsilon \end{pmatrix}$$

with a positive parameter ε which will be chosen below. Then for the interval Cholesky method we get

$$[L] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ [-1,1] & [1,\sqrt{2}] & 0 & 0 \\ 0 & [1/\sqrt{2},1] & [1,\sqrt{3/2}] & 0 \\ 0 & [2/\sqrt{2},2] & [0,1] & [\sqrt{\varepsilon},\sqrt{3+\varepsilon}] \end{pmatrix},$$

i.e., $[x]^C$ exists for any positive value of ε. On the other hand we obtain

$$[\hat{U}] = [A]^{(4)} = \begin{pmatrix} 1 & [-1,1] & 0 & 0 \\ 0 & [1,3] & 1 & 2 \\ 0 & 0 & [1,5/3] & [0,4/3] \\ 0 & 0 & 0 & [\varepsilon - 7/9, \varepsilon + 11/3] \end{pmatrix}$$

for the upper triangular matrix of the interval Gaussian algorithm. Choosing $\varepsilon = 1/3$ results in the interval $[a]^{(4)}_{44} = [-4/9, 4]$ which contains zero. Hence $[x]^G$ does not exist although $[x]^C$ does. In particular, the assumptions of Theorem 3.2 are fulfilled. Therefore, the Gaussian algorithm is feasible for any matrix $\tilde{A} \in [A]$, and our example is also a counterexample for the interval Gaussian algorithm.

[1] We thank Prof. M. Plum of the University of Karlsruhe for his suggestion to apply the symmetric part of A, which made our original proof more elementary.

The dimension $n = 4$ in Example 3.1 is minimal for a counterexample. This can be seen from our next result.

THEOREM 3.3. *Let $[A] = [A]^T \in \mathbb{IR}^{n\times n}$ contain a symmetric and positive definite matrix $\tilde{A}$, and let $n \le 3$. Then $[x]^C$ exists if and only if $[x]^G$ exists.*

Proof. By virtue of Theorem 3.1 we must only show that the existence of $[x]^C$ implies that of $[x]^G$. Therefore, from now on we assume that $[x]^C$ exists. In particular, $0 < \underline{a}_{11}$ holds, which immediately guarantees the existence of $[x]^G$ in the case $n = 1$.

$n = 2$: By Theorem 3.2 no matrix $\tilde{A} \in [A] \in \mathbb{IR}^{2\times 2}$ is singular, hence $[x]^G$ exists by Proposition 4.5.4 in [11].

$n = 3$: From $\underline{a}_{11} > 0$ we know that $\Sigma^G_{[A]}$ exists. Since any interval $[c]$ satisfies

$$[c]^2 \subseteq [c] \cdot [c] = [c]^2 + [-d, 0]$$

with an appropriate nonnegative number d, we obtain

$$\Sigma^G_{[A]} = \Sigma^C_{[A]} + [D] \quad \text{with } [D] = \operatorname{diag}([0, d_1], [0, d_2], [0, d_3]),$$

where d_1, d_2, d_3 are appropriate nonnegative real numbers. Note that $\min(\Sigma^G_{[A]})_{11} = \min(\Sigma^C_{[A]})_{11} > 0$. Choose $x \in \mathbb{R}^2\backslash\{0\}$ and $\tilde{\Sigma}^G = (\tilde{\Sigma}^G)^T \in \Sigma^G_{[A]}$. Then $\tilde{\Sigma}^G$ can be written as $\tilde{\Sigma}^G = \tilde{\Sigma}^C + \tilde{D}$ with $\tilde{\Sigma}^C = (\tilde{\Sigma}^C)^T \in \Sigma^C_{[A]}$ and $O \le \tilde{D} \in [D]$, whence

$$x^T\tilde{\Sigma}^G x = x^T\tilde{\Sigma}^C x + x^T\tilde{D}x \ge x^T\tilde{\Sigma}^C x > 0. \tag{3.2}$$

Thus any symmetric matrix $\tilde{\Sigma}^G \in \Sigma^G_{[A]}$ is positive definite, and Theorem 3.2 applies to $\Sigma^G_{[A]}$. Therefore, no matrix $\Sigma \in \Sigma^G_{[A]}$ is singular, and $[x]^G \in \mathbb{IR}^3$ exists again by virtue of Proposition 4.5.4 in [11] applied to $\Sigma^G_{[A]} \in \mathbb{IR}^{2\times 2}$. □

Another interesting negative result can be seen from Example 3.1: For symmetric and positive definite matrices $\tilde{A} \in \mathbb{R}^{n\times n}$, one proves similarly as for $\tilde{\Sigma}^G$ in (3.2) that $\tilde{A} + \tilde{D}$ with $\tilde{D} \ge O$ is positive definite, hence the Cholesky method is feasible for $\tilde{A} + \tilde{D}$, too. For interval matrices $[A] + [O, D]$, $D \ge O$, an analogous result does not hold if one merely knows that $[x]^C$ exists for $[A]$. Otherwise apply this result to $\Sigma^G_{[A]} = \Sigma^C_{[A]} + [O, D]$; it would guarantee that $\Sigma^G_{[A]}$ has a Cholesky decomposition if $[A]$ has one, and an inductive argument would show that Theorem 3.1 has a converse. This contradicts Example 3.1.

There are more classes of matrices for which one can prove the converse of Theorem 3.1. In order to characterize some of them we use the concept of an undirected graph of a real matrix $A \in \mathbb{R}^{n\times n}$ with the nodes $1, \ldots, n$ and the edges $\{i, j\}$ whenever $|a_{ij}| + |a_{ji}| \ne 0$; cf. for instance [6]. We call j a neighbor of the node i ($\ne j$) if i and j are connected by an edge. The number of neighbors of i are the degree of i in the underlying graph. Let G_k denote the kth elimination graph of $[A]$, i.e., the undirected graph of $|[A]^{(k)}|$ in which the nodes $1, \ldots, k-1$ and the corresponding edges have been removed and for which we assume that $[a]^{(k-1)}_{ij} \ne 0$ implies $[a]^{(k)}_{ij} \ne 0$, $i, j \ge k$ (no accidental zeros!); cf. [6]. If in G_k the node k has the smallest degree and if this holds for all $k = 1, \ldots, n$, then we say that $[A]$ is ordered by minimum degree. If the graph of such a matrix has tree structure (i.e., it is a connected graph with no cycles of length ≥ 3; cf. [4]), then the following result holds.

THEOREM 3.4. *Let $[A] = [A]^T \in \mathbb{IR}^{n\times n}$ contain a symmetric and positive definite matrix $\tilde{A}$. If the undirected graph of $\langle[A]\rangle$ is a tree and if it is ordered by minimum degree, then the following statements are equivalent.*

(i) *The vector* $[x]^G$ *exists.*
(ii) *The vector* $[x]^C$ *exists.*
(iii) *Each symmetric matrix in* $[A]$ *is positive definite.*

Proof. (i) $\Rightarrow$ (ii) follows from Theorem 3.1.
(ii) $\Rightarrow$ (iii) is trivial.
(iii) $\Rightarrow$ (i) follows from Theorem 3.2 and Theorem 4 in [4]. □

For a variant of the interval Cholesky method, Theorem 3.4 was proved in [4]. Note that symmetric tridiagonal interval matrices and symmetric arrowhead interval matrices [14] belong to the class of matrices characterized in Theorem 3.4 (provided that they contain a symmetric and positive definite matrix $\tilde{A}$).

Example 3.2. Consider the arrowhead matrix

$$[A] = \begin{pmatrix} 2 & 0 & [-1,1] \\ 0 & 2 & [-1,1] \\ [-1,1] & [-1,1] & 2 \end{pmatrix}.$$

Then Gerschgorin's theorem shows that the eigenvalues of each symmetric matrix $\tilde{A} \in [A]$ are nonnegative. They are even positive as can be seen in most cases by the same theorem. For the remaining cases $\tilde{A}$ is irreducibly diagonally dominant and thus an H-matrix. Since such a matrix is regular it cannot have zero as an eigenvalue. Therefore, each symmetric matrix $\tilde{A} \in [A]$ is positive definite, and $[x]^C$ exists for $[A]$ by Theorem 3.4.

In order to formulate our next result we need the extended sign matrix S' which we define recursively as in [8].

DEFINITION 3.1 (sign matrix S and extended sign matrix S' for $[A]$). *Let* $[A] \in \mathbb{IR}^{n\times n}$. *Then we have the following.*

(a) *The matrix* $S \in \mathbb{R}^{n\times n}$ *with* $s_{ij} = \operatorname{sign}\check{a}_{ij}$ *is called the sign matrix of* $[A]$.
(b) *With* S *from (a) the extended sign matrix* S' *is defined as follows:*

$S' = S$
for $k = 1 : (n-1)$
  for $i = (k+1) : n$
    for $j = (k+1) : n$
      if $s'_{ij} == 0$ *then* $s'_{ij} = -s'_{ik} s'_{kk} s'_{kj}$.

Note that the values of s'_{ij} depend only on S. Any other matrix $[\hat{A}]$ with the same sign matrix S as $[A]$ yields the same extended sign matrix S'.

THEOREM 3.5. *Let* $[A] = [A]^T \in \mathbb{IR}^{n\times n}$ *be irreducible and generalized diagonally dominant with* $0 < \underline{a}_{ii}$, $i = 1, \ldots, n$. *Moreover, let* S' *be the extended sign matrix of* $[A]$ *defined in Definition* 3.1. *Then the following statements are equivalent.*

(i) *The vector* $[x]^C$ *exists.*
(ii) *The vector* $[x]^G$ *exists.*
(iii) *The matrix* $[A]$ *is generalized irreducibly diagonally dominant or the sign condition*

$$s'_{ij}\, s'_{ik}\, s'_{kj}\, s'_{kk} = \begin{cases} 1 & \text{if } i \neq j, \\ -1 & \text{if } i = j \end{cases} \tag{3.3}$$

holds for some triple (i, j, k) *with* $k < i, j$.

(iv) *The matrix* $[A]$ *is generalized irreducibly diagonally dominant or the sign condition*

$$s'_{ij}\, s'_{ik}\, s'_{kj} = 1 \tag{3.4}$$

holds for some triple (i, j, k) *with* $k < j < i$.

Proof. The case $n = 1$ is trivial since $\underline{a}_{11} > 0$. Therefore, from now on we assume $n > 1$.

(ii) $\Leftrightarrow$ (iii) holds by virtue of Theorem 4.7 in [8].

(iii) $\Leftrightarrow$ (iv). From the general assumptions of the theorem we get $s_{ii} = 1 = s'_{ii}$, $i = 1, \ldots, n$, and $S = S^T$, whence $S' = (S')^T$. Therefore,

$$s'_{ii}\, s'_{ik}\, s'_{ki}\, s'_{kk} = (s'_{ik})^2 \neq -1$$

holds, i.e., the second sign condition in (3.3) can never be fulfilled. Moreover, a factor $s'_{kk} = 1$ can always be added in (3.4) which results in the first sign condition in (3.3). Hence the existence of some triple (i, j, k) as required in (iii) is equivalent to the existence of some triple as required in (iv).

(ii) $\Rightarrow$ (i). Since $[x]^G$ exists by assumption, each matrix $\tilde{A} \in [A]$ is regular. Consider the matrix $[A] + \varepsilon I$, $\varepsilon > 0$. Since $\underline{a}_{ii} > 0$, $i = 1, \ldots, n$, we get $\langle [A] + \varepsilon I \rangle = \langle [A] \rangle + \varepsilon I$ which shows that $[A] + \varepsilon I$ is generalized strictly diagonally dominant. Therefore, it is an H-matrix by Theorem 4.4 (a) in [8] and Theorem 2.1 guarantees that $\tilde{A} + \varepsilon I$ is positive definite for each symmetric matrix $\tilde{A} \in [A]$. Hence $\tilde{A} + \varepsilon I$ has only positive real eigenvalues which remain positive in the limit $\varepsilon \to 0$ since $\tilde{A}$ is regular and since the eigenvalues behave continuously when changing the entries of a matrix continuously. Therefore, $\tilde{A}$ is positive definite for each symmetric matrix $\tilde{A} \in [A]$. In particular, $[A]$ contains at least one such matrix, and Theorem 3.1 finishes the proof.

(i) $\Rightarrow$ (ii). Let $[x]^C$ exist and assume that $[x]^G$ does not exist. Then $[A]$ cannot be an H-matrix; in particular, by Theorem 4.4 (b) in [8] it cannot be generalized irreducibly diagonally dominant. However, since it is generalized diagonally dominant by assumption, there must exist a positive vector x such that $\langle [A] \rangle x = 0$. Without loss of generality, we can assume $x = e$, i.e.,

$$\langle [A] \rangle\, e = 0. \tag{3.5}$$

Otherwise consider the matrix $D[A]D$ with $D = \operatorname{diag}(x_1, \ldots, x_n) \in \mathbb{R}^{n \times n}$. This matrix has the same extended sign matrix S' as $[A]$, is irreducible and diagonally dominant, but not irreducibly diagonally dominant. Moreover, it fulfills (3.5), and by Lemma 2.1 the interval Cholesky method is feasible for it since it is for $[A]$ by assumption.

Since we assumed that $[x]^G$ does not exist, the equivalence of (ii) and (iii) shows that the sign condition (3.3) does not hold. Choose $k = 1$ for the moment and let S be the sign matrix of $[A]$. If $s_{ij} \neq s'_{ij}$, then s_{ij} must be zero by the construction of S' in Definition 3.1. (Note that at the beginning of this definition we have $S' = S$. Later s'_{ij} is changed only if it was equal to zero.) Therefore, $s_{ij} = s'_{ij}$ or $s_{ij} = 0$. Hence (3.3) does not hold if s'_{ij} is replaced there by s_{ij}. By Lemma 2.1 in [9] this implies

$$\left| [a]_{ij} - \frac{[a]_{i1} \cdot [a]_{1j}}{[a]_{11}} \right| = |[a]_{ij}| + \frac{|[a]_{i1}| \cdot |[a]_{1j}|}{\langle [a]_{11} \rangle} \quad \text{if } i \neq j \text{ and } i, j > 1. \tag{3.6}$$

Next we remark that the equality $|[a]^2| = |[a]|^2 = |[a] \cdot [a]|$ holds for any interval $[a]$. Since $\underline{a}_{ii}$ is positive and since $[x]^C$ exists we have $0 < \underline{l}_{ii}$ and

$$0 < (\underline{l}_{ii})^2 = \underline{a}_{ii} - \sum_{k=1}^{i-1} |[l]_{ik}^2| \le \underline{a}_{ii} - |[l]_{i1}^2| = \underline{a}_{ii} - \left| \left(\frac{[a]_{i1}}{[l]_{11}} \right)^2 \right| = \langle [a]_{ii} \rangle - \left| \frac{[a]_{i1}^2}{[a]_{11}} \right|$$

$$= \langle [a]_{ii} \rangle - \frac{|[a]_{i1}^2|}{\langle [a]_{11} \rangle} = \langle [a]_{ii} \rangle - \frac{|[a]_{i1}|^2}{\langle [a]_{11} \rangle}, \qquad i > 1.$$

In particular, $\langle [a]_{ii} \rangle > \dfrac{|[a]_{i1}|^2}{\langle [a]_{11} \rangle}$ holds, and Lemma 2.1(b) in [9] implies

$$\left\langle [a]_{ii} - \frac{[a]_{i1}^2}{[a]_{11}} \right\rangle = \left\langle [a]_{ii} - \frac{[a]_{i1} \cdot [a]_{i1}}{[a]_{11}} \right\rangle = \langle [a]_{ii} \rangle - \frac{|[a]_{i1}|^2}{\langle [a]_{11} \rangle}, \qquad i > 1. \tag{3.7}$$

From (3.6) and (3.7) we directly get

$$\langle \Sigma_{[A]}^C \rangle = \Sigma_{\langle [A] \rangle}^C = \Sigma_{\langle [A] \rangle}^G = \langle \Sigma_{[A]}^G \rangle \tag{3.8}$$

although $\Sigma_{[A]}^C \subsetneqq \Sigma_{[A]}^G$ may hold. In fact, by construction both matrices can differ at most in the diagonal because $[c]_i^2 \subsetneqq [c]_i \cdot [c]_i$ can occur. Since $[x]^C$ exists the diagonal entries of $\Sigma_{[A]}^C$ are positive; hence the sign matrices of $\Sigma_{[A]}^C$ and $\Sigma_{[A]}^G$ coincide and the same holds for the extended sign matrices.

With $e = \begin{pmatrix} 1 \\ e' \end{pmatrix}$ and (3.5) we obtain

$$\left(\langle \Sigma_{[A]}^C \rangle e' \right)_i = \left(\Sigma_{\langle [A] \rangle}^C e' \right)_i = \left(\langle [a]_{ii} \rangle - \frac{|[a]_{i1}|^2}{\langle [a]_{11} \rangle} \right) - \sum_{\substack{j=2 \\ j \ne i}}^{n} \left(|[a]_{ij}| + \frac{|[a]_{i1}| \cdot |[a]_{j1}|}{\langle [a]_{11} \rangle} \right)$$

$$= \left\{ \langle [a]_{ii} \rangle - \sum_{\substack{j=1 \\ j \ne i}}^{n} |[a]_{ij}| \right\} + \frac{|[a]_{i1}|}{\langle [a]_{11} \rangle} \left\{ \langle [a]_{11} \rangle - \sum_{j=2}^{n} |[a]_{j1}| \right\}$$

$$= \left(\langle [A] \rangle e \right)_i + \frac{|[a]_{i1}|}{\langle [a]_{11} \rangle} \left(\langle [A] \rangle e \right)_1 = 0, \qquad i = 2, \ldots, n.$$

Hence

$$\langle \Sigma_{[A]}^C \rangle e' = \Sigma_{\langle [A] \rangle}^C e' = \Sigma_{\langle [A] \rangle}^G e' = \langle \Sigma_{[A]}^G \rangle e' = 0.$$

Moreover, from (3.8) together with Lemma 3.3 in [5] we know that $\Sigma_{[A]}^C$ is irreducible provided that $n \ge 3$.

Since we assumed that $[x]^G$ does not exist, the interval Gaussian algorithm cannot be feasible for $\Sigma_{[A]}^G$. Therefore, (3.3) cannot hold when formulated for the extended sign matrix of $\Sigma_{[A]}^G$. (In fact, deleting the first row and column of S' for $[A]$ results in the corresponding extended sign matrix for the Schur complement.) Since we already showed that $\Sigma_{[A]}^C$ and $\Sigma_{[A]}^G$ have the same extended sign matrices the equivalence of (ii) and (iii) implies that the interval Gaussian algorithm is not feasible for $\Sigma_{[A]}^C$. Thus the assumptions of Theorem 3.5 for $[A]$ are also fulfilled for $\Sigma_{[A]}^C = (\Sigma_{[A]}^C)^T$. Therefore, the previous conclusions can be repeated up to the dimension $n = 2$ for

$\Sigma^C_{[A]}$. (Note that the restriction of the dimension n concerns only the irreducibility.) For ease of notation assume that $[A]$ plays the role of $\Sigma^C_{[A]}$ if $n = 2$, i.e., it is an irreducible symmetric 2×2 interval matrix satisfying $\langle [A] \rangle e = 0$. As before we obtain $\langle \Sigma^C_{[A]} \rangle e' = 0$, i.e., $0 \in \Sigma^C_{[A]} \in \mathbb{IR}^{1\times 1}$, which contradicts the feasibility of the interval Cholesky method and which finally shows that (3.3) must hold for some triple (i, j, k) unless $[A] \in \mathbb{IR}^{n\times n}$ is generalized irreducibly diagonally dominant. (In this case the sign condition (3.3) may be hurt as the example $[A] = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ shows.) □

Example 3.3.

(a) Let

$$[A]_\alpha = \begin{pmatrix} 4 & [\alpha, 2] & [\alpha, 2] \\ [\alpha, 2] & 4 & 2 \\ [\alpha, 2] & 2 & 4 \end{pmatrix}, \qquad -2 \le \alpha \le 2.$$

Then $\langle [A]_\alpha \rangle e = 0$. For $-2 < \alpha \le 2$ we obtain $S = ee^T = S'$. Thus (3.4) is fulfilled with $(i, j, k) = (3, 2, 1)$, and $[x]^C$ exists.

If $\alpha = -2$, then things change. Here

$$S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} = S'$$

and (3.4) does not hold as one can easily check. Thus $[x]^C$ does not exist. In fact, $[A]_{-2}$ contains the singular matrix

$$\tilde{A} = \begin{pmatrix} 4 & 2 & -2 \\ 2 & 4 & 2 \\ -2 & 2 & 4 \end{pmatrix}.$$

(b) Let

$$[A] = \begin{pmatrix} 4 & 0 & [0, 2] & [-2, 0] \\ 0 & 4 & [0, 2] & [0, 2] \\ [0, 2] & [0, 2] & [6, 9] & [-2, 2] \\ [-2, 0] & [0, 2] & [-2, 2] & [6, 9] \end{pmatrix}.$$

Then $[A]$ is irreducible and diagonally dominant. In particular, it satisfies the assumptions of Theorem 3.5. Moreover, we have

$$S = \begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{pmatrix} \neq S' = \begin{pmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{pmatrix}$$

with (3.4) for $(i, j, k) = (4, 3, 2)$. Hence $[x]^C$ exists.

It is easy to see by Example 3.3 (b) that (3.4) does not hold if the entries of S' are replaced there by the corresponding entries of S. Doing so, nevertheless, yields a sufficient criterion analogously to Theorem 5.3 in [5]. We state this result as a corollary which follows directly from Theorem 3.5(iv) since (3.9) below can be written as (3.4).

COROLLARY 3.1. *Let* $[A] = [A]^T \in \mathbb{IR}^{n\times n}, n \geq 3$, *be irreducible and generalized diagonally dominant with* $0 < \underline{a}_{ii}$, $i = 1, \ldots, n$. *Moreover, let* S *be the sign matrix of* $[A]$ *defined in Definition* 3.1. *If*

$$s_{ij}\, s_{ik}\, s_{kj} = 1 \tag{3.9}$$

for some triple (i, j, k) *with* $k < j < i$, *then* $[x]^C$ *exists.*

Now we consider tridiagonal matrices.

THEOREM 3.6. *Let* $[A] = [A]^T \in \mathbb{IR}^{n\times n}$ *be tridiagonal. Then the following statements are equivalent.*

(i) *The vector* $[x]^G$ *exists and* $[A]$ *contains at least one symmetric and positive definite matrix.*
(ii) *The vector* $[x]^C$ *exists.*
(iii) *Each symmetric matrix* $\tilde{A} \in [A]$ *is positive definite.*

Proof. (i) $\Rightarrow$ (ii) follows from Theorem 3.1.

(ii) $\Rightarrow$ (iii) follows from the feasibility of the Cholesky method for each symmetric matrix $\tilde{A} \in [A]$.

(iii) $\Rightarrow$ (i) follows from Theorem 2.2 and the feasibility of the interval Gaussian algorithm for H-matrices; cf. [1] or [11]. ∎

Example 3.4. Let $[A] = \text{tridiag}([-1, 1], 2, [-1, 1]) \in \mathbb{IR}^{n\times n}$. Then Gershgorin's theorem shows that the eigenvalues of each symmetric matrix $\tilde{A} \in [A]$ are nonnegative. Since $\tilde{A}$ is either irreducibly diagonally dominant or consists of blocks of such matrices, it is an H-matrix. Therefore, no eigenvalue can be zero, each symmetric matrix $\tilde{A} \in [A]$ is positive definite, and $[x]^C$ exists for $[A]$ by Theorem 3.6.

Our final result deals with matrices of the form $[A] = I + [-R, R]$, which at first glance look very specific. However, preconditioning any regular interval matrix by its midpoint inverse $\check{A}^{-1}$ finally results in such a matrix.

THEOREM 3.7. *Let* $[A] = I + [-R, R]$ *with* $O \leq R = R^T \in \mathbb{R}^{n\times n}$ *and* $0 < \underline{a}_{ii}$, $i = 1, \ldots, n$. *Then the following statements are equivalent.*

(i) *The vector* $[x]^G$ *exists.*
(ii) *The vector* $[x]^C$ *exists.*
(iii) *The spectral radius of* R *is less than one.*
(iv) *The matrix* $[A]$ *is an* H*-matrix.*

Proof. The equivalence of (i), (iii), and (iv) is contained in Theorem 3.1 of [10]; cf. also Theorem 4.2 in [8]. The implication (iv) $\Rightarrow$ (ii) follows from Theorem 2.1. For the implication (ii) $\Rightarrow$ (iv), let $[x]^C$ exist. Then the Cholesky method is feasible for $\tilde{A} = I - R = \langle [A] \rangle \in [A]$, hence $\tilde{A}$ is symmetric and positive definite. Moreover, it is an M-matrix whence $[A]$ is an H-matrix. ∎

Acknowledgment. We thank PD Dr. Uwe Schäfer of the University of Karlsruhe for carefully reading and commenting on our manuscript.

REFERENCES

[1] G. ALEFELD AND J. HERZBERGER, *Introduction to Interval Computations*, Academic Press, New York, 1983.
[2] G. ALEFELD AND G. MAYER, *The Cholesky method for interval data*, Linear Algebra Appl., 194 (1993), pp. 161–182.
[3] G. ALEFELD AND G. MAYER, *On the symmetric and unsymmetric solution set of interval systems*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1223–1240.
[4] A. FROMMER, *A feasibility result for interval Gaussian elimination relying on graph structure*, in Symbolic Algebraic Methods and Verification Methods, G. Alefeld, J. Rohn, S. Rump, and T. Yamamoto, eds., Springer, Vienna, 2001, pp. 79–86.

[5] A. Frommer and G. Mayer, *A new criterion to guarantee the feasibility of the interval Gaussian algorithm*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 408–419.
[6] A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice Hall, Englewood Cliffs, NJ, 1981.
[7] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1994.
[8] G. Mayer, *A contribution to the feasibility of the interval Gaussian algorithm*, Reliab. Comput., 12 (2006), pp. 79–98.
[9] G. Mayer and L. Pieper, *A necessary and sufficient criterion to guarantee feasibility of the interval Gaussian algorithm for a class of matrices*, Appl. Math., 38 (1993), pp. 205–220.
[10] G. Mayer and J. Rohn, *On the applicability of the interval Gaussian algorithm*, Reliab. Comput., 4 (1998), pp. 205–222.
[11] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, 1990.
[12] J. M. Ortega, *Numerical Analysis*, A Second Course, 2nd ed., SIAM, Philadelphia, 1990.
[13] K. Reichmann, *Abbruch beim Intervall–Gauss–Algorithmus*, Computing, 22 (1979), pp. 355–361.
[14] U. Schäfer, *The feasibility of the interval Gaussian algorithm for arrowhead matrices*, Reliab. Comput., 7 (2001), pp. 59–62.
[15] U. Schäfer, *Two ways to extend the Cholesky decomposition to block matrices with interval entries*, Reliab. Comput., 8 (2002), pp. 1–20.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1406–1420

CHARACTERIZING MATRICES THAT ARE CONSISTENT WITH GIVEN SOLUTIONS*

X.-W. CHANG†, C. C. PAIGE†, AND D. TITLEY-PELOQUIN†

Abstract. For given vectors $b \in \mathbb{C}^m$ and $y \in \mathbb{C}^n$ we describe a unitary transformation approach to deriving the set $\mathcal{F}$ of all matrices $F \in \mathbb{C}^{m\times n}$ such that y is an exact solution to the compatible system $Fy = b$. This is used for deriving minimal backward errors E and f such that $(A+E)y = b+f$ when possibly noisy data $A \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^m$ are given, and the aim is to decide if y is a satisfactory approximate solution to $Ax = b$. The approach might be different, but the above results are not new. However, we also prove the apparently new result that two well-known approaches to making this decision are theoretically equivalent and discuss how such knowledge can be used in designing effective stopping criteria for iterative solution techniques. All these ideas generalize to the following formulations. We extend our constructive approach to derive a superset $\mathcal{F}_{STLS+}$ of the set $\mathcal{F}_{STLS}$ of all matrices $F \in \mathbb{C}^{m\times n}$ such that y is a scaled total least squares solution to $Fy \approx b$. This is a new general result that specializes in two important ways. The ordinary least squares problem is an extreme case of the scaled total least squares problem, and we use our result to obtain the set $\mathcal{F}_{LS}$ of all matrices $F \in \mathbb{C}^{m\times n}$ such that y is an exact least squares solution to $Fy \approx b$. This complements the original less-constructive derivation of Waldén, Karlson, and Sun [*Numer. Linear Algebra Appl.*, 2 (1995), pp. 271–286]. We do the equivalent for the data least squares problem—the other extreme case of the scaled total least squares problem. Not only can the results be used as indicated above for the compatible case, but the constructive technique we use could also be applicable to other backward problems—such as those for underdetermined systems, the singular value decomposition, and the eigenproblem.

Key words. matrix characterization, approximate solutions, iterative methods, linear algebraic equations, least squares, data least squares, total least squares, scaled total least squares, backward errors, stopping criteria

AMS subject classifications. 15A06, 15A29, 65F05, 65F10, 65F20, 65F25, 65G99

DOI. 10.1137/060675691

1. Introduction.

We will study a class of "backward" problems for linear systems $Fy \approx b$. Specifically, given two vectors y and b we want to find the sets of all matrices F such that y is the exact solution (i.e., $Fy = b$), the least squares (LS) solution, the data least squares (DLS) solution, and the scaled total least square (STLS) solution. We will propose a unified unitary transformation approach to handling these problems.

Some of these problems have been investigated before. The result for the compatible case is well known, and the result for the least squares case was obtained elegantly by Waldén, Karlson, and Sun in [31]. But while [31] presents, then proves, the least squares result, our approach shows how to derive a more general result in a fairly simple way, and we suspect that this constructive approach is not only easier to comprehend for non-mathematicians, but perhaps easier to apply to other problems.

Thus the technique we use is widely applicable and an important part of this paper: it is to transform the unknown matrices F from the left and right by certain theoretical unitary matrices related to the given vectors b and y. These are designed

*Received by the editors November 27, 2006; accepted for publication (in revised form) by N. Mastronardi May 27, 2008; published electronically November 7, 2008.
http://www.siam.org/journals/simax/30-4/67569.html

†School of Computer Science, McGill University, Montreal, QC, H3A 2A7, Canada (chang@cs.mcgill.ca, paige@cs.mcgill.ca, dtitle@cs.mcgill.ca). The first author's research was supported by NSERC of Canada grant RGPIN217191-03. The second author's research was supported by NSERC of Canada grant RGPIN9236. The last author's research was supported by NSERC of Canada PGS-M Fellowship.

so that the constraints, such as $Fy = b$ in the compatible case, reveal the structure of the set of possible matrices F.

One of the main uses for finding sets of matrices consistent with given approximate solutions is to find minimal backward errors; see, for example, section 2.2 here. If the normwise relative backward error is of the order of the unit roundoff, then we say that the approximate solution is a (normwise) backward stable solution. This is useful in practice, for sometimes we do not know if an algorithm for solving a problem is numerically stable—but if we know that a computed solution of a specific problem is a backward stable solution, we are usually satisfied with this computed solution. Also, when we solve a problem by an iterative algorithm, the minimal backward error can often be used to design effective stopping criteria. There has been a lot of work on backward error problems, especially in recent years. For example, for consistent linear systems (including structured problems), see [10], [12], [19], [24], [25], [26], [30], [32]; for unconstrained least squares problems, see [6], [8], [11], [12], [13], [14], [20], [21], [22], [23], [31]; for constrained least squares problems, see [3], [13], [14]; and for data least squares problems, see [2].

To illustrate the basic ideas and techniques of our approach and the uses of the results, we start with the simplest compatible case in section 2. In section 3 we find a useful superset $\mathcal{F}_{STLS+}$ of the set $\mathcal{F}_{STLS}$ of matrices consistent with given STLS solutions. The STLS problem is a generalization of the ordinary LS and DLS problems. From the results for the STLS problem we obtain the set $\mathcal{F}_{LS}$ of consistent matrices for the LS problem and a superset $\mathcal{F}_{DLS+}$ of the set $\mathcal{F}_{DLS}$ of consistent matrices for the DLS problem. The results are given in sections 4 and 5. We have not been able to find simple and practical representations of $\mathcal{F}_{DLS}$ and $\mathcal{F}_{STLS}$, but we discuss in section 6 how the sets $\mathcal{F}_{STLS+}$ and $\mathcal{F}_{DLS+}$ can be just as useful.

In problems which have known structure we will sometimes be interested only in those matrices with that structure; see, for example, several papers in [27], [28] for total least squares problems. We have not looked at such problems, but one approach might be to start with the sets we derive here and then consider the subsets of these with the desired structures.

We will use $I = [e_1, \ldots, e_n]$ to denote the unit matrix; $\|x\|^2 \equiv x^H x$; $\|B\|_2 \equiv \sigma_{\max}(B)$, the maximum singular value of B; $\|B\|_F^2 \equiv \operatorname{trace}(B^H B)$. We will use $B^\dagger$ to represent the Moore–Penrose generalized inverse of B. For any complex vector v,

$$v^\dagger \equiv \begin{cases} 0 & \text{if } v = 0, \\ v^H/\|v\|^2 & \text{if } v \neq 0, \end{cases}$$

and $P_{v^\perp} = I - vv^\dagger$ is always the projector onto the orthogonal complement of range(v). We will regularly use the following: if $v \in \mathbb{C}^n$ and $V_2 \in \mathbb{C}^{n\times(n-1)}$, then

$$V_2 V_2^H = I - vv^\dagger \quad \Leftrightarrow \quad V \equiv [v/\|v\|, V_2] \in \mathbb{C}^{n\times n} \quad \text{is unitary.} \tag{1.1}$$

2. The compatible case. Our analysis for known results for compatible linear systems $Ax = b$ will provide the basic ideas and techniques used throughout this paper. The useful Lemma 2.1 and an apparently new result Corollary 2.3 will be given.

2.1. The set of consistent matrices F for $Fy = b$. The backward problem is the following: given $b \in \mathbb{C}^m$ and $y \in \mathbb{C}^n$ we wish to characterize all $F \in \mathbb{C}^{m\times n}$ such that y is the exact solution to $Fy = b$. We can write

$$\mathcal{F} = \mathcal{F}(b, y) \equiv \left\{ F \in \mathbb{C}^{m\times n} : \; Fy = b \right\}. \tag{2.1}$$

An explicit expression for $\mathcal{F}$ can be used in various problems such as finding optimal such F, structured such F, etc. We now obtain such an explicit characterization of all $F \in \mathcal{F}$. Note that if $y = 0$, then every $F \in \mathbb{C}^{m\times n}$ will do if $b = 0$, but none will do if $b \neq 0$, and we now consider only $y \neq 0$.

LEMMA 2.1. $b \in \mathbb{C}^m$ $y \in \mathbb{C}^n$

$$\begin{aligned} \mathcal{F} &\equiv \{F \in \mathbb{C}^{m\times n} : \ Fy = b\}, \\ \mathcal{N} &\equiv \left\{by^\dagger + Z\left(I - yy^\dagger\right) : \ Z \in \mathbb{C}^{m\times n}\right\}. \end{aligned} \tag{2.2}$$

$\mathcal{F} = \mathcal{N}$ $F \in \mathcal{F}$

$$F = by^\dagger + G_2 Y_2^H, \quad G_2 \in \mathbb{C}^{m\times(n-1)}, \quad Y_2 Y_2^H = I - yy^\dagger. \tag{2.3}$$

. In the theory of generalized inverses, any solution X of $XA = B$ can be written as $X = BA^\dagger + Z(I - AA^\dagger)$, so $\mathcal{F} = \mathcal{N}$ follows immediately. However, the following constructive derivation provides the useful representation (2.3) and can be extended to solve other backward problems such as those in sections 3, 4, and 5.

Note in (2.3) that Y_2 has $n-1$ columns, so from (1.1) we see that $Y = [y/\|y\|, Y_2]$ is unitary. Such unitary matrices are the tools we use in our constructive derivations of sets such as those above. For any $F \in \mathcal{F}$, let Y be any unitary matrix of the form $Y = [y/\|y\|, Y_2]$, so that $Y^H y = e_1\|y\|$. In order to describe our sets, we introduce an unknown matrix G. Specifically we define $G \equiv FY \in \mathbb{C}^{m\times n}$ so that the constraint $Fy = b$ can be rewritten as

$$Fy = FYY^H y = Ge_1\|y\| = b. \tag{2.4}$$

We will show how (2.4) limits the possible G. Express G as $G \equiv [g_1, G_2]$ for some vector g_1. Then (2.4) gives $g_1\|y\| = b$, so that $g_1 = b/\|y\|$ and $F = GY^H = by^\dagger + G_2 Y_2^H$, proving (2.3). Here G_2, and only G_2, is independent of the constraint $Fy = b$.

We can replace $G_2 Y_2^H$ by $ZY_2 Y_2^H$ as follows. For any $Z \in \mathbb{C}^{m\times n}$ define $G_2 \equiv ZY_2$, so that $G_2 Y_2^H = ZY_2 Y_2^H$. Conversely, for any G_2 we can define $Z \equiv G_2 Y_2^H$ so that $G_2 Y_2^H = G_2 Y_2^H Y_2 Y_2^H = ZY_2 Y_2^H$. Thus we can use (1.1) to rewrite any F in (2.3) as

$$F = by^\dagger + ZY_2 Y_2^H = by^\dagger + Z\left(I - yy^\dagger\right)$$

for some totally unknown $Z \in \mathbb{C}^{m\times n}$. Thus $F \in \mathcal{F} \Rightarrow F \in \mathcal{N}$, and so $\mathcal{F} \subseteq \mathcal{N}$. But if $F \in \mathcal{N}$, then clearly $Fy = b$, so $F \in \mathcal{F}$, proving $\mathcal{N} \subseteq \mathcal{F}$, and thus $\mathcal{F} = \mathcal{N}$. ∎

If $y = 0$ and $b = 0$, it is easy to see that $\mathcal{F} = \mathcal{N}$ still holds, but (2.3) does not since no such Y_2 can exist.

Notice that, although (2.2) is a compact explicit representation of all possible matrices F such that $Fy = b$, the equivalent (2.3) shows there are other representations. The most useful representation will depend on the problem being solved.

Compatible linear systems are a distinct and important special case of each of the later problems we examine. It is helpful to continue this introduction by applying Lemma 2.1 to give some well-known results and an interesting corollary. These illustrate how these set representations might be used in general.

2.2. Minimal backward errors and acceptable solutions. In this section we will consider only the matrix 2- and F-norms and use one description for both. Thus $\eta_{2,F}$ etc. will indicate that one can use either the matrix 2-norm throughout or the F-norm throughout.

Given $A \in \mathbb{C}^{m\times n}$, $b \in \mathbb{C}^m$, and nonzero $y \in \mathbb{C}^n$, suppose we wish to find the smallest (in some sense) perturbations E and f in A and b such that $(A+E)y = b+f$. One approach proposed by Rigal and Gaches [19, section 3.1] is to essentially solve

$$\eta_{2,F} \equiv \min_{\eta,E,f}\{\eta :\ \|E\|_{2,F} \le \eta\alpha\|A\|_{2,F},\ \|f\| \le \eta\beta\|b\|,\ (A+E)y = b+f\} \tag{2.5}$$

for given $\alpha \ge 0$ and $\beta \ge 0$ (not both zero). See Remark 2.1 for comments on $\eta_{2,F}$. Another well-known approach (see, for example, [9, Problem 7.8]) is to solve

$$\zeta \equiv \min_{E,f}\{\|[E, f\theta]\|_F :\ (A+E)y = b+f\} \tag{2.6}$$

for some given real scalar $\theta \ge 0$.

Although the solutions are known for the above two approaches, these are apparently not compared in the literature. We prove in Corollary 2.3 that for the 2- and F-norms the two approaches are, in fact, theoretically equivalent.

THEOREM 2.2. $A \in \mathbb{C}^{m\times n}$, $b \in \mathbb{C}^m$, $y \in \mathbb{C}^n$, α, β ( ), $\theta \ge 0$

$$r \equiv b - Ay, \quad \mu_{2,F} \equiv \frac{\beta\|b\|}{\alpha\|A\|_{2,F}\|y\| + \beta\|b\|}, \quad \nu \equiv \frac{1}{1+\theta^2\|y\|^2}, \tag{2.7}$$

(2.5)

$$\eta_{2,F} = \frac{\|r\|}{\alpha\|A\|_{2,F}\|y\| + \beta\|b\|}, \tag{2.8}$$

$$\hat{E} = r(1-\mu_{2,F})y^\dagger, \quad \hat{f} = -r\mu_{2,F}, \tag{2.9}$$

(2.6)

$$\zeta = \left(\frac{\theta^2\|r\|^2}{1+\theta^2\|y\|^2}\right)^{1/2}, \tag{2.10}$$

$$\hat{E} = r(1-\nu)y^\dagger, \quad \hat{f} = -r\nu. \tag{2.11}$$

. The quickest proof is to follow the approach of Higham [9, Theorem 7.1]: for each of (2.8) and (2.10) it is straightforward to show that the right-hand side is a lower bound on the minimand and that the stated optimal values give the lower bound.

But for possible future work it is useful to see how the actual solutions can be via Lemma 2.1. Using the notation of Lemma 2.1 we see from (2.1) and (2.3) that, for any given f, any E satisfying the constraint $(A+E)y = b+f$ has the form

$$E = (b+f)y^\dagger + G_2Y_2^H - A, \qquad Y_2Y_2^H = I - yy^\dagger,$$

for some $G_2 \in \mathbb{C}^{m\times(n-1)}$. Therefore with unitary Y of the form $Y = [y/\|y\|, Y_2]$,

$$\begin{aligned}\|E\|_{2,F}^2 = \|EY\|_{2,F}^2 &= \|[(b+f)y^\dagger + G_2Y_2^H - A][y/\|y\|, Y_2]\|_{2,F}^2 \\ &= \|[(b+f-Ay)/\|y\|, G_2 - AY_2]\|_{2,F}^2 \\ &\ge \|r+f\|^2/\|y\|^2.\end{aligned}$$

The last inequality becomes an equality if $G_2 = AY_2$, which is independent of f. Thus $G_2 = AY_2$ is optimal for both (2.5) and (2.6), and so with this G_2 we have $E = (b+f)y^\dagger - Ayy^\dagger = (r+f)y^\dagger$ for both problems.

Note that we can always write $f = -r\mu + u$ for some (possibly complex) scalar μ and some $u \in \mathbb{C}^m$ such that $u^H r = 0$, so

$$\|f\|^2 = \|r\|^2|\mu|^2 + \|u\|^2, \qquad \|r+f\|^2 = \|r\|^2|1-\mu|^2 + \|u\|^2.$$

But $\|E\|_{2,F} = \|r+f\|/\|y\|$, from which we can see that the minima in both (2.5) and (2.6) require $u = 0$ and μ real, since for a given real part $\mu_{\mathcal{R}}$, both $|\mu|$ and $|1-\mu|$ are minimized by taking $\mu = \mu_{\mathcal{R}}$. This gives $f = -r\mu$, $E = r(1-\mu)y^\dagger$.

The theorem is obvious when $r = 0$, so assume $r \neq 0$. For (2.5) we solve

$$\min_\mu \{\eta : |1-\mu|\cdot\|r\| \le \eta\alpha\|A\|_{2,F}\|y\|,\ |\mu|\cdot\|r\| \le \eta\beta\|b\|\},$$

from which we can see that if $\alpha = 0$, then $\mu = 1$; if $\beta = 0$, then $\mu = 0$; otherwise, the minimum occurs when

$$\eta = \frac{|1-\mu|\cdot\|r\|}{\alpha\|A\|_{2,F}\|y\|} = \frac{|\mu|\cdot\|r\|}{\beta\|b\|}, \quad 0 < \mu < 1,$$

giving $\mu = \beta\|b\|/(\alpha\|A\|_{2,F}\|y\| + \beta\|b\|) = \mu_{2,F}$ in all cases, so that the optimal $\eta = \eta_{2,F}$, proving (2.8) with its minimizers (2.9).

In (2.10)

$$\|[E, f\theta]\|_F^2 = \left[(1-\mu)^2 + \mu^2\theta^2\|y\|^2\right]\|r\|^2/\|y\|^2,$$

which is minimized by $\mu = (1+\theta^2\|y\|^2)^{-1} = \nu$, $1-\mu = \nu\theta^2\|y\|^2$, proving (2.10) with its minimizers (2.11) and completing this longer but constructive proof. ∎

Rigal and Gaches [19, section 3.1] essentially proved (2.8), while the result (2.10) is well known; see, for example, [9, Problem 7.8]. Here we relate these two results.

Corollary 2.3. [illegible] 2.2 [illegible] θ [illegible] (2.6) [illegible]

$$\theta_{2,F} \equiv \begin{cases} \left(\dfrac{\alpha\|A\|_{2,F}}{\beta\|b\|\cdot\|y\|}\right)^{1/2} & \beta > 0, \\ \infty & \beta = 0 \end{cases} \tag{2.12}$$

[illegible] $\hat{E}$ [illegible] $\hat{f}$ [illegible] (2.10) [illegible] $\hat{E}$ [illegible] $\hat{f}$ [illegible] (2.8) [illegible]. From Theorem 2.2 we see that the optimal $\hat{E}$ and $\hat{f}$ have the same forms $E = r(1-\mu)y^\dagger$ and $f = -r\mu$, where the only differences are in the values of μ. The values of μ become the same by choosing θ so that $\nu = \mu_{2,F}$; that is,

$$\nu^{-1} = 1 + \theta^2\|y\|^2 = \mu_{2,F}^{-1} = (\alpha\|A\|_{2,F}\|y\| + \beta\|b\|)/\beta\|b\|,$$

giving $\theta = \theta_{2,F}$ in (2.12) when $\beta > 0$. If $\beta = 0$, then taking $\theta = \infty$ results in $\nu = \mu_{2,F} = 0$, which forces $f = 0$; cf. the DLS case in section 5. ∎

Thus in order to define optimal backward perturbations in these cases, it does not matter which of the theoretical approaches (2.8) or (2.10) we take, as long as we choose α and β, or θ, according to (2.12).

The quantity $\eta_{2,F}$ can be used to check if an approximate solution to $Ax = b$ is an [illegible]. Most practical problems contain uncertainties in the data, and, instead of solving $Ax = b$ with ideal data A and b, we solve some system

$$(A+\delta A)\tilde{x} = b + \delta b, \quad \text{where} \quad \|\delta A\|_{2,F} \le \alpha\|A\|_{2,F},\ \|\delta b\| \le \beta\|b\| \tag{2.13}$$

for some hopefully approximately known $\alpha \geq 0$ and $\beta \geq 0$. Notice that the given y solves a problem within the range of uncertainty in the data (2.13) if and only if $\eta_{2,F} \leq 1$, so that if $\eta_{2,F} \leq 1$ we can conclude that the given y is an acceptable solution to the compatible system $Ax = b$, and this can be used as a stopping criterion for iterative methods such as the modified Gram-Schmidt GMRES method (MGS-GMRES).

2.1. If $\alpha = \beta = 1$ in (2.5), then from (2.8) $\eta_{2,F}$ becomes the normwise relative backward error (NRBE) $\|r\|/(\|A\|_{2,F}\|y\|+\|b\|)$ in [9, p. 120]. This is excellent for plotting the performance of an iterative solution of equations algorithm and can be used in the stopping criterion $\eta_{2,F} \leq O(\epsilon)$ for a numerically stable algorithm. To handle $\alpha \neq \beta$ in (2.13) we chose $\eta_{2,F}$ as in (2.5). This is then neither a NRBE nor a direct measure of backward error, and it has to be used with the very different stopping criterion $\eta_{2,F} \leq 1$. But in this case it is easy to define and compute the two NRBEs—that in A: $\eta_{2,F}\alpha = \|\hat{E}\|_{2,F}/\|A\|_{2,F}$ and that in b: $\eta_{2,F}\beta = \|\hat{f}\|/\|b\|$.

A knowledge of the uncertainties will usually suggest rough values for α and β. If we do not know such values, or want maximum accuracy in a normwise backward sense, we can use a backward stable algorithm and take $\alpha = \beta = O(\epsilon)$, where ϵ is the floating point arithmetic unit roundoff and $O(\epsilon)$ depends on the algorithm. For example, it was shown in [15, sections 1 and 8.2] that, for sufficiently nonsingular $n \times n$ A in the real problem $Ax = b$, for MGS-GMRES we would use the F-norm and might take $\alpha = \beta = 100kn\epsilon$ at step k if we wanted to be unrealistically careful, or more sensibly $\alpha = \beta = 10n\epsilon$, where experience suggests we can usually obtain even better accuracy than this.

Sometimes we will have only an estimate of the α/β ratio or of the equivalent θ satisfying (2.12). For example, we might know only that the relative error in b can be about ten times that in A. If we have no idea of the individual α and β values, we do not have a clear acceptance criterion. For certainty we could assume that α and β were very small, and in the case $\beta/\alpha \geq 1$ we could set $\alpha = O(\epsilon)$ and $\beta = (\beta/\alpha)O(\epsilon)$. For example, when using MGS-GMRES, if we know that the relative error in b is about ten times that in A, we might set $\alpha = 10n\epsilon$, $\beta = 100n\epsilon$.

If only θ, or the ratio α/β, is available, the quantity ζ in (2.6) is sometimes referred to as a normwise backward error; see, for example, [9, Problem 7.8]. But it is important to be aware that this quantity can be a poor measure of backward error for small θ. This is because $\zeta \to 0$ as $\theta \to 0$ (see (2.10)) while (2.6) shows that the optimal $E \to 0$, $f \to Ay - b = -r$ as $\theta \to 0$, so that if $r \neq 0$, then ζ will be an inappropriate measure when θ is small. A generally more appropriate measure of backward error for the $\|[E, f\theta]\|_F$ approach might be $\|[\hat{E}, \hat{f}]\|_F^2$, where with (2.11) and (2.7)

$$\left\| \left[\hat{E}, \hat{f}\right] \right\|_F^2 = \frac{1 + \theta^4\|y\|^2}{(1 + \theta^2\|y\|^2)^2}\|r\|^2. \tag{2.14}$$

Note that this quantity is equal to ζ when $\theta = 1$ and in the limit as $\theta \to \infty$ but tends to the desired $\|r\|^2$ as $\theta \to 0$. Thus, although for a given θ we minimize $\|[E, f\theta]\|_F$, a more meaningful measure of the backward error might be $\|[\hat{E}, \hat{f}]\|_F$.

Finally for fixed $\|y\|$ and $\|r\|$ the minimum of (2.14) is given by $\theta^2 = 1$. This is one argument for taking $\theta = 1$ if we have no reasonable a priori idea of α/β or θ.

3. The scaled total least squares problem. Given $A \in \mathbb{C}^{m\times n}$, $b \in \mathbb{C}^m$, and $\gamma \in (0, \infty)$, the STLS problem was formulated in [18] as finding $\hat{E}$, $\hat{f}$, and $\hat{x}$, which

solve

$$\sigma_S \equiv \min_{E,f,x} \|[E, f\gamma]\|_F \quad \text{subject to} \quad (A+E)x = b+f. \tag{3.1}$$

By taking $g = f\gamma$, (3.1) was reformulated in [16, equation (5.1)] as

$$\sigma_S \equiv \min_{E,g,x} \|[E, g]\|_F \quad \text{subject to} \quad (A+E)x\gamma = b\gamma + g. \tag{3.2}$$

The scalar σ_S is called the STLS distance, and $\hat{x} = \hat{x}(\gamma)$ the STLS solution. The formulation (3.1) is closely related to the minimal backward error problem for compatible systems (2.6), while (3.2) has the advantage of being an unscaled total least squares problem, for which codes are easily available.

Let $\mathcal{U}_{\min}$ be the left singular vector subspace of A corresponding to its minimum singular value $\sigma_{\min}(A)$. In [16] it was shown that a satisfactory condition for building the theory for the STLS problem is the condition that we will now assume holds:

$$\text{the } m \times n \text{ matrix } A \text{ has rank } n, \text{ and } b \not\perp \mathcal{U}_{\min}. \tag{3.3}$$

Under this condition the solution to (3.2) must exist and be unique.

The STLS solution reduces to the ordinary least squares solution in the limit as $\gamma \to 0$ (so $E = 0$), to the unscaled total least squares solution when $\gamma = 1$, and to the data least squares solution in the limit as $\gamma \to \infty$ (so $f = 0$); see, for example, [16].

It was shown in [4] for the real case, and in [16, equation (5.9)] for the complex case, that the STLS solution $\hat{x}$ solves

$$\sigma_S^2 = \min_x \left\{ \sigma_S^2(x) \equiv \frac{\|b - Ax\|^2}{\gamma^{-2} + \|x\|^2} \right\}. \tag{3.4}$$

If we differentiate the real version of $\sigma_S^2(x)$ in (3.4) with respect to x and equate the result to zero, we see that $\hat{x}$ satisfies the real version of

$$A^H(b - A\hat{x}) = -\hat{x}\frac{\|b - A\hat{x}\|^2}{\gamma^{-2} + \|\hat{x}\|^2} = -\hat{x}\sigma_S^2(\hat{x}).$$

This is a necessary optimality condition, but it is not sufficient since the function $\sigma_S^2(x)$ is not convex. In fact, it can be proven (see [29, Theorem 2.7], [16, section 6]) that when (3.3) holds, $\hat{x}$ solves (3.2) if and only if

$$A^H(b - A\hat{x}) = -\hat{x}\sigma_S^2, \quad \sigma_S^2 \equiv \frac{\|b - A\hat{x}\|^2}{\gamma^{-2} + \|\hat{x}\|^2} < \sigma_{\min}^2(A). \tag{3.5}$$

Given $b \in \mathbb{C}^m$ and nonzero $y \in \mathbb{C}^n$, the backward STLS problem is then to characterize the set $\mathcal{F}_{STLS}$ of all $F \in \mathbb{C}^{m\times n}$ such that y is the exact STLS solution to $Fy \approx b$; see (3.1). From (3.4) and (3.5), the sets $\mathcal{F}_{STLS}$ and $\mathcal{F}_{STLS+}$ can be defined as follows:

$$\mathcal{F}_{STLS} \equiv \left\{ F \in \mathbb{C}^{m\times n} : \ \frac{\|b - Fy\|^2}{\gamma^{-2} + \|y\|^2} = \min_{x\in\mathbb{C}^n} \frac{\|b - Fx\|^2}{\gamma^{-2} + \|x\|^2} \right\} \tag{3.6}$$

$$\equiv \left\{ F \in \mathbb{C}^{m\times n} : F^H(b - Fy) = -y\frac{\|b - Fy\|^2}{\gamma^{-2} + \|y\|^2}, \ \frac{\|b - Fy\|^2}{\gamma^{-2} + \|y\|^2} < \sigma_{\min}^2(F) \right\} \tag{3.7}$$

$$\subseteq \mathcal{F}_{STLS+} \equiv \left\{ F \in \mathbb{C}^{m\times n} : \ F^H(b - Fy) = -y\frac{\|b - Fy\|^2}{\gamma^{-2} + \|y\|^2} \right\}. \tag{3.8}$$

There might be elements of $\mathcal{F}_{STLS+}$ which do not satisfy the inequality in (3.7). In other words, there might be $F \in \mathcal{F}_{STLS+}$ for which the given y is not the minimizer, but merely a stationary point, of the right-hand side in (3.6). In the use of such sets, optimizing over $\mathcal{F}_{STLS}$ would be difficult, so in practice we would usually choose to use the more amenable $\mathcal{F}_{STLS+}$. One reason for this is that any particular element found in $\mathcal{F}_{STLS+}$ could be tested to see if it also satisfied (3.7). A more important reason is that in all the problems we can imagine we would be given a y that is a reasonable approximation to $\hat{x}$, and use this to find an $F \in \mathcal{F}_{STLS+}$ which is as close as possible to A. It can be seen from (3.5) that if $y = \hat{x}$ then the closest F in any measure would be A itself, so that F [illegible] satisfy (3.7). If we only had $y \approx \hat{x}$ then finding F as close as possible to A would tend to force (3.7) to hold. A good example of this is in [2] which deals with the minimum backward error for an approximate solution y to the DLS problem, see section 5. Using the notation in (5.4), in [2] we used

$$\mu_F(y) \equiv \min_{A+\Delta A \in \mathcal{F}_{DLS+}} \|\Delta A\|_F \quad \text{in order to find} \quad \hat{\mu}_F(y) \equiv \min_{A+\Delta A \in \mathcal{F}_{DLS}} \|\Delta A\|_F,$$

and proved in [2, Theorem 2.8] that if $\hat{x}$ is the DLS solution to $Ax \approx b$ then there exists an $\epsilon > 0$ such that if $\|y - \hat{x}\|_2 < \epsilon$ then $\mu_F(y) = \hat{\mu}_F(y)$. So that for a good approximation y nothing is lost by using $\mathcal{F}_{DLS+}$ instead of $\mathcal{F}_{DLS}$. In fact, in the thousands of numerical tests in [2, Section 5], no example was found where using $\mathcal{F}_{DLS+}$ gave the wrong answer, where the y were chosen to have relative errors up to 10^{-1}. Since $\mathcal{F}_{DLS}$ is a limiting case of $\mathcal{F}_{STLS}$, we suspect that in many practical cases $\mathcal{F}_{STLS+}$ will also be a useful and usable replacement for $\mathcal{F}_{STLS}$.

To develop an explicit expression for all $F \in \mathcal{F}_{STLS+}$, we will use the following lemma as a guide.

LEMMA 3.1. $F \in \mathbb{C}^{m\times n}$, $b \in \mathbb{C}^m$, [illegible] $y \in \mathbb{C}^n$, [illegible] $\gamma \in (0,\infty)$, [illegible]

$$F^H(b - Fy) = -y\|b - Fy\|^2 / \left(\gamma^{-2} + \|y\|^2\right) \tag{3.9}$$

$$\iff w = b - Fy, \quad \left(I - yy^\dagger\right) F^H w = 0, \quad b^H w = \frac{\|w\|^2}{1 + \gamma^2\|y\|^2} \tag{3.10}$$

$$\iff w = b - Fy, \quad \left(F^H + y\gamma^2 b^H\right) w = 0. \tag{3.11}$$

[illegible]. Define $w \equiv b - Fy$, and then

$$\|w\|^2 = b^H w - y^H F^H w. \tag{3.12}$$

Suppose that (3.9) holds. Multiplying (3.9) on the left by y^H gives

$$y^H F^H w = -\|y\|^2\|w\|^2 / \left(\gamma^{-2} + \|y\|^2\right),$$

which with (3.12) leads to the last equality in (3.10):

$$b^H w = \|w\|^2 / \left(1 + \gamma^2\|y\|^2\right). \tag{3.13}$$

The second equality in (3.10) can be obtained immediately by multiplying (3.9) on the left by $I - yy^\dagger$. Thus (3.10) holds. From (3.9) and (3.13) we obtain

$$F^H w = -y\|w\|^2 / \left(\gamma^{-2} + \|y\|^2\right) = -y\gamma^2 b^H w,$$

leading to (3.11).

Conversely if (3.10) holds, then using its second, first, and third equalities we have

$$F^H w = \frac{y\left(y^H F^H w\right)}{\|y\|^2} = \frac{y\left(b^H w - w^H w\right)}{\|y\|^2} = \frac{y\left(\frac{\|w\|^2}{1+\gamma^2\|y\|^2} - \|w\|^2\right)}{\|y\|^2} = -y\frac{\|w\|^2}{\gamma^{-2}+\|y\|^2},$$

so that (3.9) holds. Finally if (3.11) holds, then from its two equalities we obtain

$$\|w\|^2 = b^H w - y^H F^H w = b^H w - y^H\left(-y\gamma^2 b^H w\right) = \left(1+\gamma^2\|y\|^2\right) b^H w.$$

Therefore the second equality in (3.11) can be rewritten as

$$F^H w = -y\gamma^2 b^H w = -y\gamma^2\|w\|^2/\left(1+\gamma^2\|y\|^2\right),$$

so that (3.9) holds. ∎

We now obtain two new characterizations of all matrices $F \in \mathcal{F}_{STLS+}$ in (3.8).

THEOREM 3.2. $b \in \mathbb{C}^m$, $y \in \mathbb{C}^n$, $\gamma \in (0,\infty)$,

$$\mathcal{F}_{STLS+} \equiv \left\{ F \in \mathbb{C}^{m\times n} : F^H(b-Fy) = -y\frac{\|b-Fy\|^2}{\gamma^{-2}+\|y\|^2} \right\}, \tag{3.14}$$

$$\mathcal{N}_{STLS+} \equiv \left\{ (b-w)y^\dagger + \left(I-ww^\dagger\right) Z \left(I-yy^\dagger\right) : \right.$$
$$\left. w \in \mathbb{C}^m,\, b^H w = \frac{\|w\|^2}{1+\gamma^2\|y\|^2},\, Z \in \mathbb{C}^{m\times n} \right\}, \tag{3.15}$$

$$\widetilde{\mathcal{N}}_{STLS+} \equiv \left\{ -\tilde{w}\tilde{w}^\dagger b\gamma^2 y^H + \left(I-\tilde{w}\tilde{w}^\dagger\right)\left[by^\dagger + Z\left(I-yy^\dagger\right)\right] : \right.$$
$$\left. \tilde{w} \in \mathbb{C}^m, Z \in \mathbb{C}^{m\times n} \right\}. \tag{3.16}$$

$\mathcal{F}_{STLS+} = \mathcal{N}_{STLS+} = \widetilde{\mathcal{N}}_{STLS+}$, F w (3.15) $w = b - Fy$ $\tilde{w}$ (3.16) . In order to show that $\mathcal{F}_{STLS+} \subseteq \mathcal{N}_{STLS+}$ and $\mathcal{F}_{STLS+} \subseteq \widetilde{\mathcal{N}}_{STLS+}$, consider any $F \in \mathcal{F}_{STLS+}$ so that by Lemma 3.1

$$w \equiv b - Fy, \quad \left(F^H + y\gamma^2 b^H\right) w = 0. \tag{3.17}$$

If $w = 0$, then $Fy = b$, so from Lemma 2.1 we see that $F \in \mathcal{F} = \mathcal{N}$. But obviously $\mathcal{N} \subseteq \mathcal{N}_{STLS+}$ and $\mathcal{N} \subseteq \widetilde{\mathcal{N}}_{STLS+}$; thus $F \in \mathcal{N}_{STLS+}$ and $F \in \widetilde{\mathcal{N}}_{STLS+}$. Now assume $w \neq 0$. Write $\hat{w} \equiv w/\|w\|$, $\hat{y} \equiv y/\|y\|$, and let $W = [\hat{w}, W_2] \in \mathbb{C}^{m\times m}$ and $Y = [\hat{y}, Y_2] \in \mathbb{C}^{n\times n}$ be unitary matrices, so that $Y^H y = e_1\|y\|$ and $W^H w = e_1\|w\|$. Define $G \equiv W^H F Y \in \mathbb{C}^{m\times n}$, so $F = WGY^H$. The restrictions in (3.17) can then be written as

$$Ge_1\|y\| = W^H(b-w), \quad G^H e_1\|w\| = -e_1\left(\|y\|\gamma^2 b^H w\right). \tag{3.18}$$

We will now show how (3.18) limits the possible G. Write

$$m \times n \;\; G = \begin{bmatrix} g_{11} & g_1^H \\ g_2 & G_{22} \end{bmatrix}, \;\; \text{with} \;\; (m-1)\times(n-1) \;\; G_{22}.$$

Then from (3.18) we obtain

$$\begin{bmatrix} g_{11} \\ g_2 \end{bmatrix} \|y\| = \begin{bmatrix} \hat{w}^H(b-w) \\ W_2^H b \end{bmatrix}, \quad \begin{bmatrix} g_{11}^H \\ g_1 \end{bmatrix} \|w\| = \begin{bmatrix} -\|y\|\gamma^2 b^H w \\ 0 \end{bmatrix},$$

leading to

$$g_1 = 0, \quad g_2 = W_2^H b/\|y\|, \quad g_{11} = \hat{w}^H(b-w)/\|y\| = -\|y\|\gamma^2 \hat{w}^H b. \tag{3.19}$$

From these it follows that

$$\begin{aligned} F = WGY^H &= (\hat{w}g_{11} + W_2 g_2)\hat{y}^H + W_2 G_{22} Y_2^H \\ &= ww^\dagger(b-w)y^\dagger + W_2 W_2^H b y^\dagger + W_2 G_{22} Y_2^H \qquad (3.20) \\ &= -ww^\dagger b\gamma^2 y^H + W_2 W_2^H b y^\dagger + W_2 G_{22} Y_2^H. \qquad (3.21) \end{aligned}$$

Similarly to what we did in the proof of Lemma 2.1, we can replace $W_2 G_{22} Y_2^H$ in (3.20) and (3.21) by $W_2 W_2^H Z Y_2 Y_2^H$ for some totally unknown $Z \in \mathbb{C}^{m\times n}$. Then using (1.1) we have from (3.20) and (3.21) that

$$\begin{aligned} F &= ww^\dagger(b-w)y^\dagger + \left(I - ww^\dagger\right) b y^\dagger + \left(I - ww^\dagger\right) Z \left(I - yy^\dagger\right) \\ &= (b-w)y^\dagger + \left(I - ww^\dagger\right) Z \left(I - yy^\dagger\right) \qquad (3.22) \\ &= -ww^\dagger b\gamma^2 y^H + \left(I - ww^\dagger\right)\left[b y^\dagger + Z\left(I - yy^\dagger\right)\right]. \qquad (3.23) \end{aligned}$$

From (3.22) and (3.10) it follows that $F \in \mathcal{N}_{STLS+}$, so $\mathcal{F}_{STLS+} \subseteq \mathcal{N}_{STLS+}$; and from (3.23) it follows that $F \in \widetilde{\mathcal{N}}_{STLS+}$, and therefore $\mathcal{F}_{STLS+} \subseteq \widetilde{\mathcal{N}}_{STLS+}$.

Conversely suppose that $F \in \mathcal{N}_{STLS+}$, so that

$$F = (b-w)y^\dagger + \left(I - ww^\dagger\right) Z \left(I - yy^\dagger\right)$$

for some Z and some w satisfying $b^H w = \|w\|^2/(1+\gamma^2\|y\|^2)$. Then it follows that

$$Fy = b - w, \quad \left(I - yy^\dagger\right) F^H w = 0. \tag{3.24}$$

Therefore by Lemma 3.1 and (3.14), $F \in \mathcal{F}_{STLS+}$, and thus $\mathcal{N}_{STLS+} \subseteq \mathcal{F}_{STLS+}$, proving that $\mathcal{N}_{STLS+} = \mathcal{F}_{STLS+}$. Finally suppose that $F \in \widetilde{\mathcal{N}}_{STLS+}$, so that

$$F = -\tilde{w}\tilde{w}^\dagger b\gamma^2 y^H + \left(I - \tilde{w}\tilde{w}^\dagger\right)\left[b y^\dagger + Z\left(I - yy^\dagger\right)\right]$$

for some Z and $\tilde{w}$. Then $\tilde{w}^H F = -\tilde{w}^H b\gamma^2 y^H$, so $(F^H + y\gamma^2 b^H)\tilde{w} = 0$, and

$$Fy = -\tilde{w}\tilde{w}^\dagger b\gamma^2\|y\|^2 + \left(I - \tilde{w}\tilde{w}^\dagger\right) b = b - \tilde{w}\tilde{w}^\dagger b\left(1 + \gamma^2\|y\|^2\right).$$

This gives an expression for w defined by

$$w \equiv b - Fy = \tilde{w}\left[\tilde{w}^\dagger b\left(1 + \gamma^2\|y\|^2\right)\right]. \tag{3.25}$$

We see by Lemma 3.1 and (3.14) that $F \in \mathcal{F}_{STLS+}$, and thus $\widetilde{\mathcal{N}}_{STLS+} \subseteq \mathcal{F}_{STLS+}$, proving that $\mathcal{F}_{STLS+} = \widetilde{\mathcal{N}}_{STLS+}$. The first equality in (3.24) and (3.25) indicates that w in (3.15) is a scalar multiple of $\tilde{w}$ in (3.16) for the same matrix F. ∎

Here we make two remarks. Unlike the expression for $\mathcal{N}_{STLS+}$ in (3.15), the expression for $\widetilde{\mathcal{N}}_{STLS+}$ in (3.16) does not involve any constraint and so is easier to use. But if we want to consider $\gamma \to \infty$, it is easier to use $\mathcal{N}_{STLS+}$; see section 5.

The condition (3.3) does not necessarily hold for every $F \in \mathcal{F}_{STLS+}$—an example is the rank-1 matrix $by^\dagger \in \mathcal{F}_{STLS+}$ in (3.8) which does not have full column rank if $n > 1$ nor need it hold for every $F \in \mathcal{F}_{STLS}$. This knowledge needs to be taken into account in the use of these sets, but at least we know that, for every $F \in \mathcal{F}_{STLS+}$, y gives a stationary point of $\|b - Fx\|^2/(\gamma^{-2} + \|x\|^2)$; see (3.4).

This completes our theory for the general STLS formulation. We will now use Theorem 3.2 to characterize matrices consistent with given approximate solutions for its two extreme cases: the least squares and data least squares problems.

4. The least squares problem. Given $A \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^m$, the ordinary LS problem is defined as

$$\sigma_{LS} \equiv \min_{f,x} \|f\| \quad \text{subject to} \quad Ax = b + f.$$

It is well known that $\hat{x}$ is the LS solution if and only if it satisfies the normal equations:

$$A^H(b - A\hat{x}) = 0.$$

See, for example, [1] or [5] for useful background.

Given $b \in \mathbb{C}^m$ and nonzero $y \in \mathbb{C}^n$, the backward LS problem is then to characterize the set $\mathcal{F}_{LS}$ of all $F \in \mathbb{C}^{m\times n}$ such that y is the exact LS solution to $Fy \approx b$. Obviously we have

$$\begin{aligned}\mathcal{F}_{LS} &\equiv \left\{F \in \mathbb{C}^{m\times n} : \|b - Fy\|^2 = \min_{x\in\mathbb{C}^n} \|b - Fx\|^2\right\} \\ &= \left\{F \in \mathbb{C}^{m\times n} : F^H(b - Fy) = 0\right\}.\end{aligned}$$

We now give an alternative derivation to that in [31] of an explicit representation for all $F \in \mathcal{F}_{LS}$.

THEOREM 4.1. [illegible] $b \in \mathbb{C}^m$ [illegible] $y \in \mathbb{C}^n$, [illegible]

$$\mathcal{F}_{LS} \equiv \left\{F \in \mathbb{C}^{m\times n} : \ \|b - Fy\|^2 = \min_{x\in\mathbb{C}^n} \|b - Fx\|^2\right\}$$

$$= \left\{F \in \mathbb{C}^{m\times n} : \ F^H(b - Fy) = 0\right\}, \tag{4.1}$$

$$\widetilde{\mathcal{N}}_{LS} \equiv \left\{\left(I - \tilde{w}\tilde{w}^\dagger\right)\left[by^\dagger + Z\left(I - yy^\dagger\right)\right] : \ \tilde{w} \in \mathbb{C}^m, Z \in \mathbb{C}^{m\times n}\right\}. \tag{4.2}$$

[illegible] $\mathcal{F}_{LS} = \widetilde{\mathcal{N}}_{LS}$ [illegible] F [illegible] $w \equiv b - Fy$ [illegible] $\tilde{w}$ [illegible] (4.2) [illegible]. This theorem could be proved by the same approach as that used in proving Theorem 3.2. But we can obtain the results directly from Theorem 3.2. Notice from (3.14) and (4.1) that

$$\mathcal{F}_{LS} = \lim_{\gamma\to 0} \mathcal{F}_{STLS+}.$$

Now since $\mathcal{F}_{STLS+} = \widetilde{\mathcal{N}}_{STLS+}$ from Theorem 3.2, it follows that

$$\begin{aligned}\mathcal{F}_{LS} &= \lim_{\gamma\to 0} \mathcal{F}_{STLS+} = \lim_{\gamma\to 0} \widetilde{\mathcal{N}}_{STLS+} \\ &= \left\{\left(I - \tilde{w}\tilde{w}^\dagger\right)\left[by^\dagger + Z\left(I - yy^\dagger\right)\right] : \tilde{w} \in \mathbb{C}^m, Z \in \mathbb{C}^{m\times n}\right\} = \widetilde{\mathcal{N}}_{LS}.\end{aligned}$$

The conclusion that $w = b - Fy$ is a scalar multiple of $\tilde{w}$ still holds. In fact, this can be seen from (3.25) by taking $\gamma \to 0$. ∎

In [31] Waldén, Karlson, and Sun gave the original and elegant proof that $\mathcal{F}_{LS} = \widetilde{\mathcal{N}}_{LS}$, and used the result to find the minimal backward error for the LS problem. It is not the intent of this paper to find minimal backward errors, and we will now specialize the general result of Theorem 3.2 to DLS problems.

5. The data least squares problem. Given $A \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^m$, the DLS problem is defined as (see [7] and, for example, [16], [17]):

$$\sigma_{D} \equiv \min_{E,x} \|E\|_F \quad \text{subject to} \quad (A+E)x = b. \tag{5.1}$$

When $\gamma \to \infty$, the STLS problem (3.1) becomes the DLS problem (5.1); see [16]. The condition (3.3) is still needed for building the theory for the DLS problem.

It is easy to show that (5.1) is equivalent to (see, e.g., [16])

$$\sigma_{D}^2 = \min_{x} \frac{\|b-Ax\|^2}{\|x\|^2}. \tag{5.2}$$

From [16, equations (5.14)–(5.17)], when (3.3) holds, $\hat{x}$ solves (5.1) if and only if

$$A^H(b-A\hat{x}) = -\hat{x}\sigma_{D}^2, \quad \sigma_{D}^2 \equiv \frac{\|b-A\hat{x}\|^2}{\|\hat{x}\|^2} < \sigma_{\min}^2(A). \tag{5.3}$$

Both (5.2) and (5.3) can also be obtained by taking $\gamma \to \infty$ in (3.4) and (3.5).

Given $b \in \mathbb{C}^m$ and nonzero $y \in \mathbb{C}^n$, the backward DLS problem is then to characterize the set $\mathcal{F}_{DLS}$ of all $F \in \mathbb{C}^{m\times n}$ such that y is the exact DLS solution to $Fy \approx b$. As in the STLS problem, the sets $\mathcal{F}_{DLS}$ and $\mathcal{F}_{DLS+}$ can be defined as follows:

$$\begin{aligned} \mathcal{F}_{DLS} &\equiv \left\{ F \in \mathbb{C}^{m\times n} : \ \frac{\|b-Fy\|^2}{\|y\|^2} = \min_{x\in\mathbb{C}^n} \frac{\|b-Fx\|^2}{\|x\|^2} \right\} \\ &\subseteq \mathcal{F}_{DLS+} \equiv \left\{ F \in \mathbb{C}^{m\times n} : \ F^H(b-Fy) = -y\frac{\|b-Fy\|^2}{\|y\|^2} \right\}. \end{aligned} \tag{5.4}$$

Comments paralleling those given after (3.8) and Theorem 3.2 apply here as well.

We now obtain an explicit characterization for all $F \in \mathcal{F}_{DLS+}$.

THEOREM 5.1. [illegible] $b \in \mathbb{C}^m$ [illegible] $y \in \mathbb{C}^n$, [illegible]

$$\mathcal{F}_{DLS+} \equiv \left\{ F \in \mathbb{C}^{m\times n} : \ F^H(b-Fy) = -y\frac{\|b-Fy\|^2}{\|y\|^2} \right\},$$

$$\mathcal{N}_{DLS+} \equiv \left\{ (b-w)y^\dagger + \left(I-ww^\dagger\right) Z \left(I-yy^\dagger\right) : w \in \mathbb{C}^m,\, b^Hw = 0,\, Z \in \mathbb{C}^{m\times n} \right\}. \tag{5.5}$$

[illegible] $\mathcal{F}_{DLS+} = \mathcal{N}_{DLS+}$, [illegible] $F \in \mathcal{N}_{DLS+}$, [illegible] w [illegible] (5.5) [illegible] $w = b - Fy$ [illegible]. We could prove this theorem by using a constructive derivation similar to that used in proving Theorem 3.2. Instead we obtain the results by taking the limit $\gamma \to \infty$ for the results in Theorem 3.2. In fact, we have

$$\begin{aligned} \mathcal{F}_{DLS+} &= \lim_{\gamma\to\infty} \mathcal{F}_{STLS+} = \lim_{\gamma\to\infty} \mathcal{N}_{STLS+} \\ &= \left\{ (b-w)y^\dagger + \left(I-ww^\dagger\right) Z \left(I-yy^\dagger\right) : \ b^Hw = 0,\ Z \in \mathbb{C}^{m\times n} \right\} \\ &= \mathcal{N}_{DLS+}. \end{aligned}$$

The conclusion that w in (5.5) satisfies $w = b - Fy$ still holds and can also be verified by forming Fy for any $F \in \mathcal{N}_{DLS+}$. ∎

The result of Theorem 5.1 is used in [2] for the backward perturbation analysis for the DLS problem.

6. Summary and comments. Given $b \in \mathbb{C}^m$ and $y \in \mathbb{C}^n$ we have presented a unitary transformation approach to finding sets, or supersets, of all matrices $F \in \mathbb{C}^{m\times n}$ such that y is the solution to $Fy \approx b$ for some common classes of approximation problems.

Our approach is constructive and easy to follow. We have used the well-known compatible case $Fy = b$ to illustrate this approach in its simplest setting, as well as to illustrate one of the uses of such sets—finding minimal backward errors. In doing so we have shown the equivalence of two often used problem formulations for such errors—an apparently new result.

We then applied this approach to finding new and useful supersets of matrices consistent with the STLS solution to $Fy \approx b$. From (3.1) or (3.5) the STLS solution becomes the LS solution as $\gamma \to 0$ and becomes the DLS solution as $\gamma \to \infty$; see, for example, [16, section 6]. Based on these facts, we derived the results for the LS and DLS problems using the results of Theorem 3.2 directly, although we could have separately given a full constructive derivation for these two problems, similar to that in Theorem 3.2.

We summarize the different problems and sets we have obtained as follows:

- the STLS problem; see (3.8) and Theorem 3.2:

$$\begin{aligned}
\mathcal{F}_{STLS} &\equiv \left\{ F \in \mathbb{C}^{m\times n} : \frac{\|b-Fy\|^2}{\gamma^{-2}+\|y\|^2} = \min_{x\in\mathbb{C}^n} \frac{\|b-Fx\|^2}{\gamma^{-2}+\|x\|^2} \right\} \\
\subseteq \mathcal{F}_{STLS+} &\equiv \left\{ F \in \mathbb{C}^{m\times n} : F^H(b-Fy) = -y\frac{\|b-Fy\|^2}{\gamma^{-2}+\|y\|^2} \right\} \\
= \mathcal{N}_{STLS+} &\equiv \Big\{ (b-w)y^\dagger + \left(I-ww^\dagger\right) Z \left(I-yy^\dagger\right) : \\
&\qquad w \in \mathbb{C}^m,\ w^H b = \frac{\|w\|^2}{1+\gamma^2\|y\|^2},\ Z \in \mathbb{C}^{m\times n} \Big\} \\
= \widetilde{\mathcal{N}}_{STLS+} &\equiv \left\{ -\tilde{w}\tilde{w}^\dagger b\gamma^2 y^H + \left(I-\tilde{w}\tilde{w}^\dagger\right)\left[by^\dagger + Z\left(I-yy^\dagger\right)\right] : \right. \\
&\qquad \left. \tilde{w} \in \mathbb{C}^m, Z \in \mathbb{C}^{m\times n} \right\}.
\end{aligned}$$

- compatible systems; see Lemma 2.1:

$$\begin{aligned}
\mathcal{F} &\equiv \{F \in \mathbb{C}^{m\times n} : Fy = b\} \\
= \mathcal{N} &\equiv \left\{ by^\dagger + Z\left(I-yy^\dagger\right) :\ Z \in \mathbb{C}^{m\times n} \right\}.
\end{aligned}$$

- the LS problem; see Theorem 4.1:

$$\begin{aligned}
\mathcal{F}_{LS} &\equiv \{F \in \mathbb{C}^{m\times n} : \|b-Fy\| = \min_{x\in\mathbb{C}^n} \|b-Fx\|\} \\
&= \{F \in \mathbb{C}^{m\times n} :\ F^H(b-Fy) = 0\} \\
= \widetilde{\mathcal{N}}_{LS} &\equiv \left\{ \left(I-\tilde{w}\tilde{w}^\dagger\right)\left[by^\dagger + Z\left(I-yy^\dagger\right)\right] :\ \tilde{w} \in \mathbb{C}^m,\ Z \in \mathbb{C}^{m\times n} \right\}.
\end{aligned}$$

- the DLS problem; see Theorem 5.1:

$$\begin{aligned}
\mathcal{F}_{DLS} &\equiv \left\{ F \in \mathbb{C}^{m\times n} : \frac{\|b-Fy\|^2}{\|y\|^2} = \min_{x\in\mathbb{C}^n} \frac{\|b-Fx\|^2}{\|x\|^2} \right\} \\
\subseteq \mathcal{F}_{DLS+} &\equiv \left\{ F \in \mathbb{C}^{m\times n} : F^H(b-Fy) = -y\frac{\|b-Fy\|^2}{\|y\|^2} \right\} \\
= \mathcal{N}_{DLS+} &\equiv \left\{ (b-w)y^\dagger + \left(I-ww^\dagger\right) Z \left(I-yy^\dagger\right) : b^H w = 0, w \in \mathbb{C}^m, Z \in \mathbb{C}^{m\times n} \right\}.
\end{aligned}$$

The sets $\mathcal{F}_{STLS+}$ and $\mathcal{F}_{DLS+}$ are supersets of $\mathcal{F}_{STLS}$ and $\mathcal{F}_{DLS}$, respectively. But some theoretical arguments and numerical experiments given in [2] have shown that, when y is a reasonable approximation to the solution of the DLS problem for $Ax \approx b$, the set $\mathcal{N}_{DLS+}$ can usually be used with no further constraints to obtain the minimal backward errors for the DLS problem. This is probably true for the STLS problem as well. However, since such behavior is problem-dependent, we will not discuss it further here, except to state that for many practical uses $\mathcal{N}_{STLS+}$ or $\widetilde{\mathcal{N}}_{STLS+}$ can be used in place of $\mathcal{F}_{STLS}$, and $\mathcal{N}_{DLS+}$ can be used in place of $\mathcal{F}_{DLS}$.

The constructive technique we use could also be applicable to other backward problems, e.g., finding a matrix whose partial eigenvalues and eigenvectors are known.

Acknowledgment. We would like to thank the referees for their helpful comments.

REFERENCES

[1] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[2] X.-W. Chang, G. H. Golub, and C. C. Paige, *Backward Perturbation Analysis for Data Least Squares*, manuscript.
[3] A. J. Cox and N. J. Higham, *Backward error bounds for constrained least squares problems*, BIT, 39 (1999), pp. 210–227.
[4] G. H. Golub and C. F. Van Loan, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.
[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
[6] J. F. Grcar, *Optimal Sensitivity Analysis of Linear Least Squares*, Technical report LBNL-52434, Lawrence Berkeley National Laboratory, 2003.
[7] R. D. D. Groat and E. M. Dowling, *The data least squares problem and channel equalization*, IEEE Trans. Signal Process., 42 (1993), pp. 407–411.
[8] M. Gu, *Backward perturbation bounds for linear least squares problems*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 363–372.
[9] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[10] D. J. Higham and N. J. Higham, *Backward error and condition of structured linear systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 162–175.
[11] R. Karlson and B. Waldén, *Estimation backward perturbation bounds for the linear least squares problem*, BIT, 37 (1997), pp. 862–869.
[12] D. S. Mackey, N. Mackey, and F. Tisseur, *Structured Mapping Problems for Matrices Associated with Scalar Products Part* I: *Lie and Jordan Algebras*, Technical report, Manchester Institute for Mathematical Sciences, 2006.
[13] A. N. Malyshev, *Optimal backward perturbation bounds for the LSS problems*, BIT, 41 (2001), pp. 430–432.
[14] A. N. Malyshev and M. Sadkane, *Computation of optimal backward perturbation bounds for large sparse linear least squares problems*, BIT, 41 (2002), pp. 739–747.
[15] C. C. Paige, M. Rozložník, and Z. Strakoš, *Modified Gram-Schmidt (MGS), Least Squares, and Backward Stability of MGS-GMRES*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 264–284.
[16] C. C. Paige and Z. Strakoš, *Scaled total least squares fundamentals*, Numer. Math., 91 (2002), pp. 117–146.
[17] C. C. Paige and Z. Strakoš, *Unifying least squares, total least squares and data least squares*, in Total Least Squares and Errors-in-Variables Modeling, S. Van Huffel and P. Lemmerling, eds., Kluwer Academic, Dordrecht, The Netherlands, 2002, pp. 25–34.
[18] B. D. Rao, *Unified treatment of LS, TLS and truncated SVD methods using a weighted TLS framework*, in Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modelling, S. Van Huffel, ed., SIAM, Philadelphia, 1997, pp. 11–20.
[19] J. L. Rigal and J. Gaches, *On the compatibility of a given solution with the data of a linear system*, J. ACM, 14 (1967), pp. 543–548.
[20] G. W. Stewart, *On the perturbation of pseudo-inverses, projections, and linear least squares problems*, SIAM Rev., 19 (1977), pp. 634–662.

[21] Z. Su, *Computational Methods for Least Squares Problems and Clinical Trials*, Ph.D. thesis, Scientific Computing & Computational Mathematics, Stanford University, Palo Alto, CA, 2005.

[22] J.-G. Sun, *Optimal backward perturbation bounds for the linear least squares problem with multiple right-hand sides*, IMA J. Numer. Anal., 16 (1996), pp. 1–11.

[23] J.-G. Sun, *On optimal backward perturbation bounds for the linear least-squares problem*, BIT, 37 (1997), pp. 179–188.

[24] J.-G. Sun, *Bounds for the structured backward errors of Vandermonde systems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 45–59.

[25] J.-G. Sun, *A note on backward errors for structured linear systems*, Numerical Linear Algebra with Applications, 12(7), 2005, pp. 585-603.

[26] J.-G. Sun and Z. Sun, *Optimal backward perturbation bounds for underdetermined systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 393–402.

[27] S. Van Huffel, ed., *Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling*, SIAM, Philadelphia, 1997.

[28] S. Van Huffel and P. Lemmerling, eds., *Total Least Squares and Errors-in-Variables Modeling*, Kluwer Academic, Dordrecht, The Netherlands, 2002.

[29] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM, Philadelphia, 1991.

[30] J. M. Varah, *Backward error estimates for Toeplitz systems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 408–417.

[31] B. Waldén, R. Karlson, and J. Sun, *Optimal backward perturbation bounds for the linear least squares problem*, Numer. Linear Algebra Appl., 2 (1995), pp. 271–286.

[32] H. Xiang and Y. Wei *On normwise structured backward errors for saddle point systems*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 838-849.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1421–1423

A NOTE ON EIGENVALUES OF MATRICES WHICH ARE SELF-ADJOINT IN SYMMETRIC BILINEAR FORMS*

ZHI-HAO CAO†

Abstract. In this note we discuss the eigenvalue properties of matrices which are self-adjoint in symmetric bilinear forms and point out that an assertion given by Stoll and Wathen in [*SIAM J. Matrix Anal. Appl.*, 30 (2008), pp. 582–608] is not true.

Key words. eigenvalues, symmetric bilinear forms, symmetric indefinite matrices

AMS subject classifications. 65F10, 65F15

DOI. 10.1137/080728068

Let $\mathcal{H} \in \mathcal{R}^{n,n}$ be a symmetric matrix. The symmetric bilinear form $< \cdot, \cdot >_{\mathcal{H}}$ defined by $\mathcal{H}$ is

$$< x, y >_{\mathcal{H}} :\equiv \ x^T \mathcal{H} y.$$

The fact that a matrix $\mathcal{A} \in \mathcal{R}^{n,n}$ is self-adjoint in the symmetric bilinear form $< \cdot, \cdot >_{\mathcal{H}}$ means that

$$x^T \mathcal{A}^T \mathcal{H} y =< \mathcal{A}x, y >_{\mathcal{H}} =< x, \mathcal{A}y >_{\mathcal{H}} = x^T \mathcal{H}\mathcal{A} y \, \forall x, y \in \mathcal{R}^n,$$

which is equivalent to

$$\mathcal{A}^T \mathcal{H} = \mathcal{H}\mathcal{A}; \tag{1}$$

i.e., $\mathcal{H}\mathcal{A}$ is symmetric. Thus, the fact that the matrix $\mathcal{A}$ is self-adjoint in the symmetric bilinear form $< \cdot, \cdot >_{\mathcal{H}}$ can also be defined by (1), and we say $\mathcal{A}$ is $\mathcal{H}$ [illegible].

It is well known that if $\mathcal{A}$ is I-symmetric, i.e., $\mathcal{A}^T = \mathcal{A}$, then $\mathcal{A}$ has real eigenvalues. In [3, p. 587], Stoll and Wathen discuss eigenvalues of matrices which are self-adjoint in symmetric bilinear forms, as follows.

Assume that a matrix $\mathcal{A} \in \mathcal{R}^{n,n}$ is $\mathcal{H}$ [illegible], where $\mathcal{H} \in \mathcal{R}^{n,n}$ is symmetric; i.e., $\mathcal{A}^T \mathcal{H} = \mathcal{H}\mathcal{A}$ holds and (λ, v) is a given eigenpair of $\mathcal{A}$. Thus

$$\mathcal{A}v = \lambda v, \quad v \neq 0. \tag{2}$$

Multiplying (2) from the left by $v^* \mathcal{H}$, where v^* is the conjugate transpose of v, gives

$$v^* \mathcal{H}\mathcal{A} v = \lambda v^* \mathcal{H} v. \tag{3}$$

From (3), Stoll and Wathen claim that λ must be real. Thus, they [3, p. 587, lines 29–30] state: "Note that the above arguments establish that there is no symmetric bilinear form in which $\mathcal{A}$ is self-adjoint unless $\mathcal{A}$ has real eigenvalues."

However, the assertion above is not true as the following example shows.

*Received by the editors June 23, 2008; accepted for publication (in revised form) by H. A. van der Vorst August 9, 2008; published electronically November 7, 2008. This work is supported by NSFC Project 10871051.

http://www.siam.org/journals/simax/30-4/72806.html

†School of Mathematical Sciences and Laboratory of Mathematics for Nonlinear Sciences, Fudan University, Shanghai 200433, People's Republic of China (zcao@fudan.edu.cn, zhcao@cableplus.com.cn).

EXAMPLE.

$$\mathcal{A} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \qquad \mathcal{H} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

$\mathcal{A}$ $\pm i$ $\mathcal{H}$

$$\mathcal{H}\mathcal{A} \equiv \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \equiv \mathcal{A}^T\mathcal{H}.$$

$\mathcal{A}$ $< \cdot, \cdot >_{\mathcal{H}}$

(λ, v) $\mathcal{A}$ $\lambda = i$ $v^* = [1, i]$

(2)

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ -i \end{pmatrix} = i \begin{pmatrix} 1 \\ -i \end{pmatrix}.$$

$v^*\mathcal{H}$ (3)

$$v^*\mathcal{H}\mathcal{A}v \equiv [1, i] \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ -i \end{pmatrix} = 0 = i[1, i] \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ -i \end{pmatrix} \equiv \lambda v^*\mathcal{H}v.$$

λ

We can revise the assertion of Stoll and Wathen as follows.

PROPOSITION A1. $\mathcal{A} \in \mathcal{R}^{n,n}$ $\mathcal{H}$-symmetric $\mathcal{H} \in \mathcal{R}^{n,n}$ $v^*\mathcal{H}v \neq 0$ $\mathcal{A}$ $\mathcal{A}$ $\mathcal{H}$ $\mathcal{A}$ $\mathcal{H}$ $\mathcal{A}$

It should be noted that Benzi and Simoncini in [1] investigated spectral properties of the block 2×2 matrix of the form

$$\mathcal{M}_- = \begin{pmatrix} A & B^T \\ -B & C \end{pmatrix},$$

where $A \in \mathcal{R}^{n,n}$ is symmetric positive definite, $B \in \mathcal{R}^{m,n}$ with $m \leq n$, and $C \in \mathcal{R}^{m,m}$ is symmetric positive semidefinite. Let

$$\mathcal{J} = \begin{pmatrix} I_n & \\ & -I_m \end{pmatrix},$$

which is symmetric indefinite. Then it is easy to see that

$$\mathcal{J}\mathcal{M}_- = \mathcal{M}_-^T\mathcal{J};$$

i.e., $\mathcal{M}_-$ is $\mathcal{J}$. The eigenvalues of the matrix $\mathcal{M}_-$ are usually not all real (cf. [1, section 2]).

Finally, we point out that an important related result is given by Gohberg, Lancaster, and Rodman (cf. [2, Corollary 5.2]).

PROPOSITION A2. $\mathcal{A}$ $\mathcal{H}$-symmetric $\mathcal{H}$ $\mathcal{H}$ $\mathcal{A} = \mathcal{H}^{-1}\mathcal{A}^T\mathcal{H}$

Comment from Stoll and Wathen. This short note by Professor Cao gives an important correction to some incorrect statements in our SIAM Journal on Matrix Analysis and Applications paper

Martin Stoll and Andy Wathen,

SIAM J. Matrix Anal. Appl., 30 (2008), pp. 582–608.

Precisely on page 587, the one-sentence paragraph before Lemma 3.8 should say "Note that the above arguments establish that there is no inner product in which $\mathcal{A}$ is self-adjoint unless $\mathcal{A}$ has real eigenvalues," the preceding paragraph should have the ammended sentence "On the right-hand side $v^*\mathcal{H}v$ is also real since $\mathcal{H}$ is real symmetric, therefore the eigenvalue must be real unless $v^*\mathcal{H}v = 0$," and the final sentence in that paragraph is not relevant.

The case when $v^*\mathcal{H}v = 0$ for some eigenvector v of $\mathcal{A}$ is more important than we had realized and Professor Cao significantly points out that for real square matrix $\mathcal{A}$ there is a real symmetric matrix $\mathcal{H}$ such that $\mathcal{H}\mathcal{A} = \mathcal{A}^T\mathcal{H}$; i.e., that for real square matrix there is a symmetric bilinear form in which it is self-adjoint.

These corrections do not affect any of the other results and statements in our paper as far as we are aware.

Martin Stoll, Andy Wathen
Oxford University Computing Laboratory
7th August 2008

REFERENCES

[1] M. Benzi and V. Simoncini, *On the eigenvalues of a class of saddle point matrices*, Numer. Math., 103 (2006), pp. 173–196.
[2] I. Gohberg, P. Lancaster, and L. Rodman, *Matrices and Indefinite Scalar Products*, Birkhäuser-Verlag, Basel, 1983.
[3] M. Stoll and A. Wathen, *Combination preconditioning and the Bramble–Pasciak$^+$ preconditioner*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 582–608.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1424–1444

FAST COMPUTATION OF MINIMAL FILL INSIDE A GIVEN ELIMINATION ORDERING*

PINAR HEGGERNES† AND BARRY W. PEYTON‡

Abstract. Minimal elimination orderings were introduced by Rose, Tarjan, and Lueker in 1976, and during the last decade they have received increasing attention. Such orderings have important applications in several different fields, and they were first studied in connection with minimizing fill in sparse matrix computations. Rather than computing any minimal ordering, which might result in fill that is far from minimum, it is more desirable for practical applications to start from an ordering produced by a fill-reducing heuristic and then compute a minimal fill that is a subset of the fill produced by the given heuristic. This problem has been addressed previously, and there are several algorithms for solving it. The drawback of these algorithms is that either there is no theoretical bound given on their running time, although they might run fast in practice, or they have a good theoretical running time, but they have never been implemented, or they require a large machinery of complicated data structures to achieve the good theoretical time bound. In this paper, we present an algorithm called MCS-ETree for solving the mentioned problem in $O(nm\,\mathrm{A}(m,n))$ time, where m and n are, respectively, the number of edges and vertices of the graph corresponding to the input sparse matrix and $\mathrm{A}(m,n)$ is the very slowly growing inverse of Ackerman's function. A primary strength of MCS-ETree is its simplicity and its straightforward implementation details. We present run time test results to show that our algorithm is fast in practice. Thus our algorithm is the first that both has a provably good running time with easy implementation details and is fast in practice.

Key words. sparse matrix computations, minimal fill, elimination trees, composite tree rotations, maximum cardinality search (MCS), minimal triangulation

AMS subject classifications. 65F05, 65F50, 05C85, 05C90

DOI. 10.1137/070680680

1. Introduction.

Consider the Cholesky factorization $A = LL^T$ of an $n \times n$ symmetric positive definite sparse matrix A. Elements $l_{ij} \neq 0$, where $a_{ij} = 0$, are called *fill* elements. It is well known that finding a good permutation matrix P and computing the Cholesky factor of PAP^T rather than the Cholesky factor of A can give much less fill and is an essential operation in sparse matrix computations. The matrix A is conveniently interpreted as a graph G, where G has a vertex v_i for each row (or equivalently column) i of A and $\{v_i, v_j\}$ is an edge in G if and only if $a_{ij} \neq 0$. Similarly, the *filled graph* G^+ is the graph of $L + L^T$, and the fill elements of L correspond to the *fill edges* of G^+. Any permutation matrix P for A corresponds to an elimination ordering α on G such that G_α is the graph of PAP^T, and the number of fill edges in G_α^+ is entirely dependent on α. Thus we refer to the fill edges of G_α^+ as the fill *produced by* α. (Definitions and notation are detailed in section 2.)

For sparse matrix computations [26] and in many other fields [7, 16, 27], one would like to find orderings that produce the minimum possible fill. This problem was shown to be NP-hard by Yannakakis in 1981 [28]. Already in 1976, Rose, Tarjan, and Lueker [25] conjectured the NP-hardness of this problem. They also introduced the notion of *minimal elimination orderings* and *minimal fill*, and they presented an algorithm for computing both in $O(nm)$ time in the same paper. An ordering α is a minimal elimination ordering if there is no ordering β such that G_β^+ is a strict subgraph

*Received by the editors January 22, 2007; accepted for publication (in revised form) by E. Ng July 17, 2008; published electronically December 3, 2008.
http://www.siam.org/journals/simax/30-4/68068.html
†Department of Informatics, University of Bergen, N-5020 Bergen, Norway (pinar@ii.uib.no).
‡Dalton State College, Dalton, GA 30720 (bpeyton@daltonstate.edu).

of G_α^+. For any ordering α, the filled graph G_α^+ is a chordal graph [10] and is called a *triangulation* of G. If α is a minimal elimination ordering, then G_α^+ is a minimal triangulation. One reason that minimal elimination orderings are highly desirable in sparse matrix computations is that they ensure that subsequent equivalent reorderings do not change the space allocation requirements [5]. In the field of graph algorithms, minimal elimination orderings and minimal triangulations are very important and well studied [13], as they include the set of triangulations that correspond to widely studied graph parameters, like *minimum fill* and *treewidth*, and thus provide a tool to compute these by approximation algorithms [21] or exact (fast) exponential time algorithms [9].

A minimal triangulation can contain fill that is far from minimum fill. Consequently, for practical applications, it is more appropriate to start with a good triangulation produced by a common heuristic algorithm, like Minimum Degree [1, 17] or Nested Dissection [11], and then compute a minimal triangulation that is a subgraph of the initial triangulation [6]. This problem, sometimes called the *minimal triangulation sandwich problem*, was first posed and solved by Blair, Heggernes, and Telle in 1996 [5], and they presented an algorithm with running time $O(mf + f^2)$, where f is the number of fill edges in the initial triangulation. For small f, this algorithm is fast in practice; however, its running time is heavily dependent on f, which might be $O(n^2)$, giving an $O(n^4)$ time algorithm in the worst case. Later, Dahlhaus solved the same problem with an algorithm of running time $O(nm)$ [8], but this algorithm has never been implemented to our knowledge. A more recent algorithm by Berry et al. solves the same problem in $O(nm)$ time [3]; however, a heavy machinery of complicated data structures is necessary to achieve this time bound. In addition to these, two algorithms based on iterations were given without running time analysis separately by Peyton [24] and by Berry, Heggernes, and Simonet [4]. The algorithm of Peyton is documented to run fast in practice,[1] whereas the latter algorithm is of less practical and more theoretical interest [22].

In this paper, we present an algorithm called MCS-ETree that takes as input a graph G and an initial ordering β and produces as output a minimal elimination ordering α such that G_α^+ is a subgraph of G_β^+ (i.e., G_α^+ is *sandwiched* between G and G_β^+). The running time of our algorithm is $O(nm\,\mathrm{A}(m,n))$, where $\mathrm{A}(m,n)$ is the very slowly growing inverse of Ackerman's function. Hence, our theoretical running time is very close to the best known theoretical running time $O(nm)$ for solving this problem. Compared to $O(nm)$ algorithms solving the same problem, MCS-ETree has the advantage of being both fast in practice and easy to implement, while not relying on complicated data structures; it uses basic operations and data structures commonly used in practice in sparse matrix computations, with modest adaptations for use by the algorithm. In addition, in practical tests our algorithm is usually faster than the previous algorithm with the fastest running time.

This paper is organized as follows. Section 2 introduces most of the background, terminology, and notation. Section 3 gives some background on composite elimination tree rotations [19], which are used by our new algorithm in a slightly modified form. Section 4 presents the new algorithm MCS-ETree, which computes minimal orderings and solves the above-mentioned sandwich problem. This section also proves that the algorithm is correct. Section 5 discusses some of the implementation issues and shows that the running time is $O(nm\,\mathrm{A}(m,n))$. Also, section 5 both presents a straightforward implementation and discusses how to enhance the implementation in

[1] In fact, the algorithm of Peyton [24] is the fastest in practice of all mentioned algorithms.

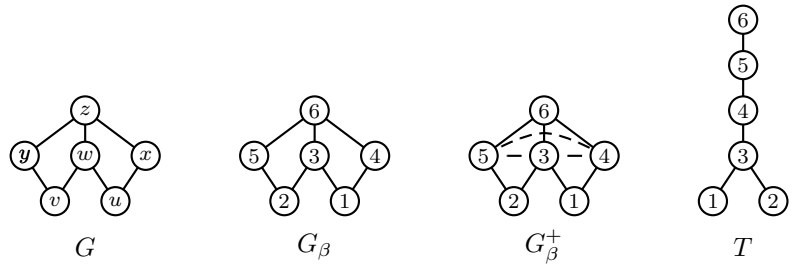


FIG. 2.1. *For the given graph G and its ordering β, the Elimination Game results in the filled graph G_β^+. The three fill edges are dashed lines. Also pictured is the elimination tree produced by the ordering.*

ways that dramatically improve the performance in our tests. Section 6 reports the results of these tests. Finally, section 7 gives some concluding remarks.

2. Background and notation. A graph $G = (V, E)$ consists of a set V of n vertices and a set E of m edges. For a given graph G, we denote the set of its vertices by $V(G)$ and the set of its edges by $E(G)$. When $\{u, v\}$ is an edge, we say that u and v are *adjacent* or *neighbors*. For a vertex v of G, $\mathrm{adj}_G(v)$ denotes the set of vertices adjacent to v, also called the *adjacency* or *neighborhood* of v, and $\mathrm{adj}_G[v] = \mathrm{adj}_G(v) \cup \{v\}$. For a set of vertices $X \subseteq V$, $\mathrm{adj}_G(X) = \bigcup_{x \in X} \mathrm{adj}_G(x) \setminus X$ and $\mathrm{adj}_G[X] = \mathrm{adj}_G(X) \cup X$. An *ordering* α of G is a bijective function $\alpha : V \to \{1, 2, \ldots, n\}$, and we will sometimes write $\alpha = (v_1, v_2, \ldots, v_n)$, meaning that $\alpha(v_i) = i$ for $1 \le i \le n$ (we will call i the *number* of v_i). When an ordering α is given, the ordered graph is denoted by G_α. In this case, $\mathrm{hadj}_{G_\alpha}(v_i) = \mathrm{adj}_{G_\alpha}(v_i) \cap \{v_{i+1}, v_{i+2}, \ldots, v_n\}$ is the set of higher-numbered neighbors of v_i, and $\mathrm{ladj}_{G_\alpha}(v_i) = \mathrm{adj}_{G_\alpha}(v_i) \cap \{v_1, v_2, \ldots, v_{i-1}\}$ is the set of lower-numbered neighbors of v_i.

If two graphs G and H have the same vertex set, then G is a *subgraph* of H if $E(G) \subseteq E(H)$, and G is a *proper subgraph* of H if $E(G) \subset E(H)$. A subgraph of G *induced* by a vertex set $X \subseteq V$ will be denoted by $G(X)$. An induced subgraph $G(X)$ contains every edge of G with both endpoints in X. A set $X \subseteq V$ of vertices is a *clique* if every pair of vertices in X is adjacent in G. A *maximal clique* is any clique whose vertex set is maximal with respect to subset inclusion for this property. A *path* is a sequence of distinct vertices $x_1 - x_2 - \cdots - x_k$ such that x_i is adjacent to x_{i+1} for $1 \le i < k$. A *chord* on a path is an edge between two nonconsecutive vertices of the path. A *cycle* is a path where the first vertex is the same as the last vertex. A graph is *chordal* if it contains no chordless cycle on four or more vertices.

The following simple algorithm is called the *Elimination Game* [23], and it simulates (on graphs) the Cholesky factorization of matrices. With input graph G and ordering α, repeatedly pick the smallest numbered vertex, add edges to make its set of neighbors a clique, and remove this vertex and the edges incident upon it from the graph until the graph is empty. The set of edges that are added during the algorithm is called *fill*, and the *filled graph* G_α^+ is obtained by adding to G this fill. Figure 2.1 shows a graph G, an ordering β of the graph (G_β), and the filled graph G_β^+ associated with the ordering. This graph and initial ordering will be used throughout the paper to illustrate algorithms and other points as needed.

The following lemma, which we will use in our proofs, characterizes the edges in a filled graph G_α^+.

LEMMA 2.1 (see [25]). *Given a graph G and an ordering $\alpha = (v_1, v_2, \ldots, v_n)$, an edge $\{v_i, v_j\}$, with $i < j$ is present in G_α^+ if and only if $\{v_i, v_j\}$ is an edge of G or there is a path between v_i and v_j in G containing only vertices from the set $\{v_1, v_2, \ldots, v_{i-1}\}$.*

A path between v_i and v_j containing only vertices that all have smaller numbers in α than the smaller of i and j will be called a *fill path*.

If G_α^+ contains no fill edges, then α is called a *perfect elimination ordering (peo)*. Fulkerson and Gross showed that chordal graphs are exactly the class of graphs that have perfect elimination orderings [10]. Thus for every graph G and ordering α, the filled graph G_α^+ is chordal, and G_α^+ is a triangulation of G. In a chordal graph, the vertices of any maximal clique can be ordered last by some peo in any arbitrary internal order [27].

Any ordering β of G that is a peo of G_α^+ is an *equivalent reordering* of G with respect to α. An equivalent reordering introduces no new fill, that is, G_β^+ is a subgraph of G_α^+. If there exists no ordering β for which G_β^+ is a proper subgraph of G_α^+, then α is called a *minimal elimination ordering (meo)*, and G_α^+ is a *minimal triangulation*. Computing an meo is equivalent to computing a minimal triangulation, as every peo of a minimal triangulation gives the same filled graph when applied to the original graph [5, 25]. The following is a characterization of minimal triangulations that we will use in the proof that our new algorithm is correct.

THEOREM 2.2 (see [25]). *A given triangulation H of a graph G is a minimal triangulation if and only if every fill edge added to G to obtain H is the unique chord of a 4-cycle in H.*

Given a graph G and an ordering α, the filled graph G_α^+ defines a structure called an *elimination tree* T as follows: vertex v_j is the parent of vertex v_i in T if v_j is the smallest numbered vertex in $\mathrm{hadj}_{G_\alpha^+}(v_i)$. The elimination tree associated with the filled graph in Figure 2.1 is included in Figure 2.1. Due to Lemma 2.1, the elimination tree corresponding to α can be computed directly from G and α without computing G_α^+ explicitly [20]. A *topological ordering* of T is any ordering that numbers each child with a number smaller than that of its parent. The ordering in Figure 2.1 is a topological ordering of T. Any topological ordering of T is an equivalent ordering of G with respect to α. Consequently, we will talk about equivalent orderings with respect to an ordering and with respect to an elimination tree interchangeably. Liu [20] provides a thorough examination of elimination trees. Note also that if G has more than one connected component, then one obtains an elimination forest with one tree for each connected component.

In a rooted tree, an *ancestor* of a vertex v is any vertex that is on the unique path between v and the root, including the root; a *descendant* of v is any vertex of which v is an ancestor. Let $T[v]$ be the subtree of an elimination tree T that is rooted at v and consists of v and every descendant of v in T; such subtrees will be called *elimination subtrees*. It is well known that $\mathrm{hadj}_{G_\alpha^+}(v) = \mathrm{adj}_G(V(T[v]))$ [20], and we will make use of this fact throughout the paper. As an illustration, note that in Figure 2.1 we have

$$\begin{aligned}\mathrm{adj}_G(V(T[v_3])) &= \mathrm{adj}_G(\{v_1, v_2, v_3\})\\ &= \{v_4, v_5, v_6\}\\ &= \mathrm{adj}_{G_\beta^+}(v_3).\end{aligned}$$

Finally, we let $\mathrm{anc}_T(v)$ be the set of ancestors of v in T, where v is *not* included in the set; we also write $\mathrm{anc}_T[v] = \mathrm{anc}_T(v) \cup \{v\}$.

Algorithm Change_Root(G, T, u)
Input: A graph $G = (V, E)$, an elimination tree T of G, and a vertex $u \in V$.
Output: A reordering γ of G that is equivalent with respect to T,
where u is numbered last.

```
Number u last in γ and mark u as already numbered;
z ← u;
while z is not the root of T do
    Order the unnumbered vertices of adj_G(V(T[z])) last in γ, but
    before those that are already numbered by γ;
    Mark the newly numbered vertices as already numbered;
    z ← the parent of z in T;
end while;
Number in γ the vertices in V \ anc_T[u] using their original
relative order in T;
```

end Change_Root;

FIG. 3.1. *An algorithm for changing the root of an elimination tree with an equivalent reordering (see Algorithm* 3.2 *Composite_Rotations in Liu* [19]).

3. Changing the root of an elimination subtree. In our new algorithm MCS-ETree, we will need to reorder an elimination subtree $T[v]$ in such a way that a particular vertex $u \in V(T[v])$ is numbered last by this reordering, and the corresponding reordering of $G(V(T[v]))$ is equivalent to any given topological ordering of $T[v]$. Since u is numbered last among the vertices in $V(T[v])$ by the reordering, it will be the root of the new elimination subtree associated with the new equivalent reordering. A trivial modification of the composite elimination tree rotations algorithm in Liu [19] will perform this task.

For a given graph G and a given ordering β, let T be the elimination tree associated with the filled graph G^+_β, and let $u \in V(G)$. Algorithm 3.2 (Composite_Rotations) from [19] reorders G with a peo γ of G^+_β such that the vertices of $\mathrm{adj}_G(V(T[u]))$ are numbered last in γ. (Recall that γ is an equivalent reordering of G with respect to β.) Notice that there might be ancestors of u in T that do not belong to $\mathrm{adj}_G(V(T[u]))$, and hence u will often become closer to the root of the resulting new elimination tree corresponding to γ and will never be further away than it is in T. The algorithm Change_Root in Figure 3.1 adds a single first line to Liu's Composite_Rotations algorithm in order to number u last and also modifies the last line to number the vertices in $V \setminus \mathrm{anc}_T[u]$ so that u is not also numbered there. These are the only modifications to the original algorithm. Consequently, the rest of the vertices are numbered in the same order as in Composite_Rotations, that is, the vertices of $\mathrm{adj}_G(V(T[u]))$ are ordered next-to-last and so on. The elimination tree obtained from the ordering produced by Change_Root clearly is rooted at vertex u.

In Figure 3.2, we illustrate a run of Change_Root on the graph and elimination tree in Figure 2.1. The elimination tree in Figure 3.2 is the same as that in Figure 2.1 with the numbers replaced by the appropriate letters. The vertex u has been chosen to become the new root. The algorithm first numbers u last with the number 6. The main loop then processes the ancestors of u in ascending order. When u is processed, the unnumbered neighbors of $V(T[u])$, namely, w and x, are numbered next-to-last in front of u with numbers 4 and 5, respectively. When w is processed next, the unnumbered neighbors of $V(T[w])$, namely, z and y, are numbered last in front of the previously numbered vertices with numbers 2 and 3, respectively. When y and

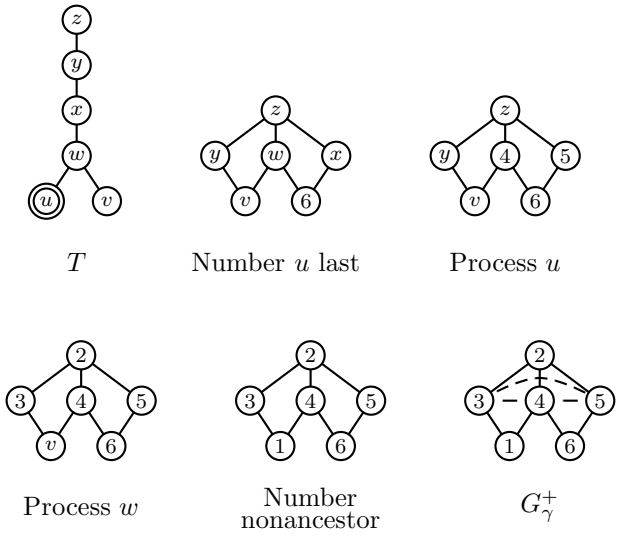


FIG. 3.2. *The algorithm Change_Root is run on the graph and elimination tree shown in Figure* 2.1. *The new root is to be vertex u.*

z are processed, no vertices receive their numbers. Finally, when the main loop is finished, the single nonancestor of u, namely v, receives the number 1. Note that G^+_γ in Figure 3.2 is identical to G^+_β in Figure 2.1 so that γ is an equivalent reordering with respect to β, as required. Note also that u will be the root of the new elimination tree, as desired.

The following lemma shows that Change_Root produces an ordering equivalent to the input ordering.

LEMMA 3.1. *Let β be an ordering of a graph G, and let T be the elimination tree associated with G^+_β. Choose $u \in V(G)$. Any ordering γ produced by Change_Root(G, T, u) is equivalent with respect to β.*

Proof. The desired property is inherited directly from Composite_Rotations, as we will see. Consider the ordering γ' produced by Composite_Rotations that corresponds as closely as possible to the ordering γ produced by Change_Root. It is known from Liu [19] that γ' is equivalent with respect to β. The ordering γ' matches the ordering γ except for the placement of u. The ordering γ numbers u at the end of the ordering; the ordering γ' numbers u before the vertices of $\mathrm{adj}_G(V(T[u]))$ and after the vertices of $V(T[u]) \setminus \{u\}$. The lowest numbered vertex of $\mathrm{adj}_G(V(T[u]))$ will be the parent of u in the elimination tree T' associated with γ'. Note then that the set $\{u\} \cup \mathrm{adj}_G(V(T[u]))$ is a clique in $G^+_{\gamma'}$ and forms a chain in T' from u to the root of the tree. A new topological ordering of T' can be obtained by ordering the vertices of $\{u\} \cup \mathrm{adj}_G(V(T[u]))$ last (consistent with γ') and then ordering the rest earlier, but consistent with γ'. The key observation is that when we change this ordering so that u is moved to the end (i.e., u is ordered last, but the vertices of $\mathrm{adj}_G(V(T[u]))$ remain ordered consistent with γ'), we obtain a topological ordering of the elimination tree associated with the ordering γ. Since we have merely changed the order of the vertices in the highest-numbered clique, clearly, γ is equivalent with respect to γ'. Since γ' is equivalent with respect to β, we have γ is equivalent with respect to β, as desired. □

We will use Change_Root to process elimination subtrees. Choose a vertex $v \in V(T)$ and consider the elimination subtree $T[v]$. Observe that $T[v]$ is the elimination

tree one obtains by applying a topological ordering of $T[v]$ to the induced subgraph $H = G(V(T[v]))$. This follows because any topological ordering of an elimination tree will produce the same elimination tree. Let $u \in V(T[v])$. When we execute Change_Root(H, $T[v]$, u), we obtain an equivalent reordering of H with respect to $T[v]$, where u is numbered last and hence will become the root of the new elimination subtree. When this new subtree is glued to the old elimination tree with the old parent of v now becoming the new parent of u, a revised elimination tree for the entire graph is obtained.

4. Algorithm MCS-ETree and its proof of correctness. In this section, we present a new algorithm MCS-ETree that solves the minimal triangulation sandwich problem. Given an original ordering β and graph G, this algorithm generates a minimal ordering α such that G_α^+ is a subgraph of G_β^+. The algorithm generates α by numbering the vertices from n down to 1, and the key feature is the selection at each step of a vertex of "maximum cardinality" to receive the next number. Hence the algorithm resembles a minimal triangulation algorithm called MCS-M [2], and its proof of correctness uses the same technique used there. In section 5, we will show that it can be implemented to run in $O(nm\,\mathrm{A}(m,n))$ time and can be implemented so that it does not compute any filled graphs explicitly.

Our new algorithm is given in Figure 4.1. First the algorithm computes the elimination tree T^* obtained when β is used as an elimination order on G. The set of elimination subtrees remaining to be processed (i.e., numbered) is $Trees$. Initially $Trees$ contains the single member $T^* = T^*[x]$, where x is the root of T^*. We have assumed that G is a connected graph so that we have an elimination tree rather than an elimination forest. If we had an elimination forest, then we would place each tree of the forest in $Trees$.

4.1. Executing MCS-ETree on an example. Figure 4.2 walks step-by-step through an execution of MCS-ETree on the example introduced in Figure 2.1. In Figure 4.2, the filled graph G_β^+ is the same as that pictured in Figure 2.1. The initialization step computes the elimination tree T^* of G with respect to β. This is shown next in the figure. The elimination subtree $T^*[x]$ is placed in $Trees$, the set of elimination subtrees yet to receive their α-numbers. The set of vertices that have received their α-numbers, namely L, is initially empty. The set L is listed above the heavy horizontal line above T^*. The counter k used to assign the next α-number is initially 6.

The algorithm then enters the main *while* loop. There is only one unnumbered elimination subtree to process, which is referred to as $T = T[v]$ by the algorithm. Now, $T = T^*$ at this stage in the example. Since $L = \emptyset$, the cardinalities referred to next in the algorithm are all zero. In the figure, these zero cardinalities are written beside each vertex. A vertex of maximum cardinality, with no descendant of maximum cardinality, will have to be a leaf in this case. The algorithm chooses vertex 1 as the maximum cardinality vertex.

The algorithm next uses Change_Root to reorder the subtree and change the root to vertex 1. Precisely this operation was illustrated in Figure 3.2. The ordering shown in the first graph of Figure 4.2, labeled G_γ^+, is precisely the same as the final ordering shown in Figure 3.2.

The algorithm next computes the elimination subtree for the new reordering. This new elimination tree T' rooted at the vertex now numbered 6 by γ appears next in the figure. At the bottom of the *while* loop, the root vertex receives its α-number 6. It is removed and added to the set of α-numbered vertices L. The unnumbered elimination

Algorithm MCS-ETree
Input: A graph G and an ordering β.
Output: An meo α of G such that G_α^+ is a subgraph of G_β^+.

```
/* Initializations */
Compute the elimination tree T* of G with respect to β;
x ← root of T*;
Trees ← {T*[x]};   L ← ∅;   k ← n;

while Trees ≠ ∅ do

    /* Get an unnumbered elimination subtree T */
    Pick an arbitrary elimination subtree T = T[v] from Trees;
    Trees ← Trees \ {T};

    /* Find a special vertex u in T of "maximum cardinality" */
    Find a vertex u ∈ V(T) for which |adj_G(V(T[u])) ∩ L| = |adj_G(V(T))|,
    and |adj_G(V(T[w])) ∩ L| < |adj_G(V(T[u])) ∩ L| for each descendant w of
    u in T;

    /* Reorder the subtree and change the root to u */
    H ← G(V(T));
    Compute a topological order γ1 of T;
    Use Change_Root(H, T, u) to compute a peo γ2 of H+_γ1 that numbers u last;

    /* Compute the elimination subtree for the new reordering */
    Compute the elimination tree T' = T'[u] of H with respect to γ2;

    /* Number u and store the unnumbered subtrees for future processing */
    for each child c of u in T' do
        Trees ← Trees ∪ {T'[c]};
    end for;
    α(u) ← k;   L ← L ∪ {u};   k ← k − 1;

end while;

end MCS-ETree;
```

FIG. 4.1. *Algorithm MCS-ETree, which finds a minimal ordering and solves the minimal triangulation sandwich problem.*

subtree rooted at the sole child of vertex 6 (i.e., vertex 5) is also added to $Trees$ to be processed later.

At the top of the next iteration of the *while* loop, the unnumbered subtree $T'[c]$, where c is the vertex numbered 5 by the current ordering γ, is chosen to be processed next. Note that in the previously filled graph, only vertices 5 and 4 are neighbors of the vertex that received α-number 6. The cardinalities of vertices 4 and 5 then are both 1, while the cardinalities of the rest of the vertices in the subtree are 0, as indicated in the figure. The vertex with γ-number 4 is the maximum cardinality vertex with no descendants of maximum cardinality.

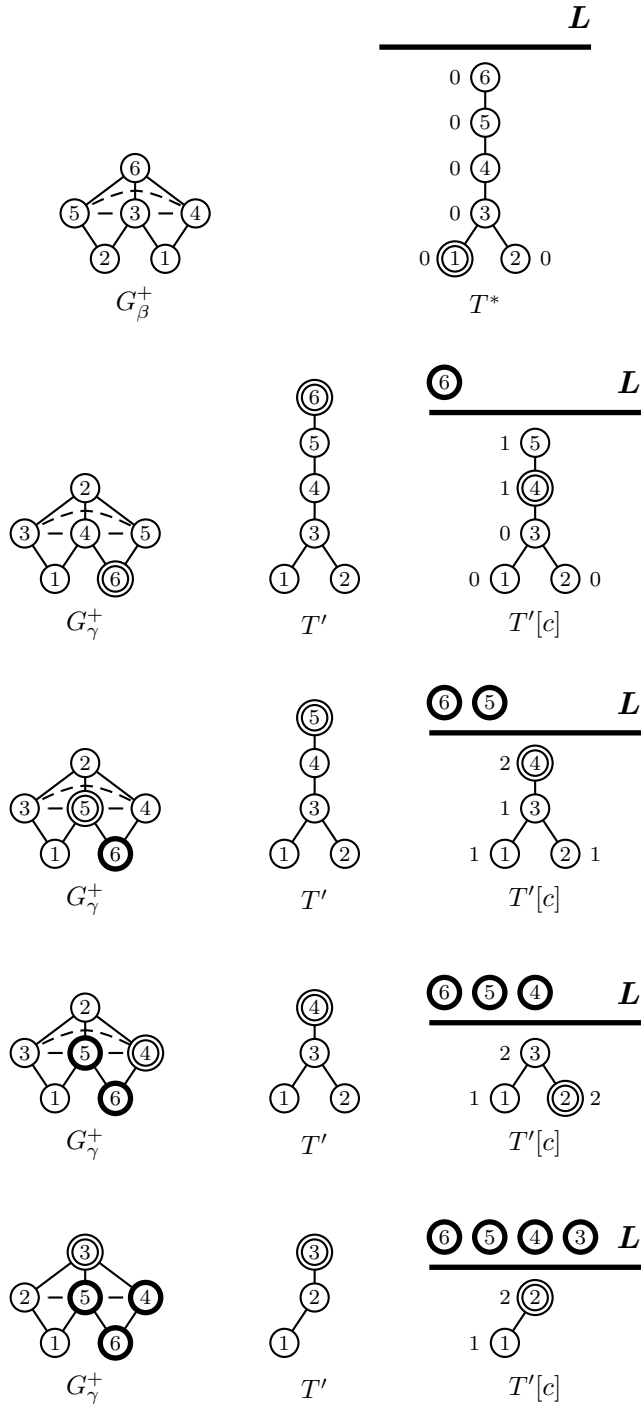


FIG. 4.2. *An execution of Algorithm MCS-ETree on the example introduced in Figure* 2.1.

We leave it for the reader to walk through the remaining steps of MCS-ETree on our example shown in Figure 4.2. Observe that in the final filled graph G^+_γ, the fill edge that once joined current vertices 2 and 4 has disappeared. In the unfilled graph, vertices 1, 5, 3, and 2 form an unchorded 4-cycle and so do vertices 6, 4, 3, and 5 (using the current γ-numbers). No single fill edge will suffice to chord both cycles, so any minimum fill set will have two fill edges—one to chord each cycle. It follows that the last filled graph pictured has minimum fill, and hence minimal fill. We leave it for the reader to verify that the final ordering α produced by the algorithm is the same as that shown in the last graph G^+_γ shown at the bottom of the figure. Hence the final ordering α is a minimal ordering.

4.2. Proving the algorithm correct. In this subsection, we will show that the algorithm is correct. The following simple loop invariant is needed to verify the integrity of the elimination subtrees.

LEMMA 4.1. *The following is a loop invariant of algorithm MCS-ETree:*

$$adj_G(V(T)) \subseteq L \text{ for each elimination subtree } T \in Trees.$$

Proof. The statement is clearly true before the first iteration of the *while* loop. Suppose that it is true at the beginning of an iteration. Then, for the elimination subtree $T = T[v]$ chosen to be processed and removed from $Trees$, we have $\mathrm{adj}_G(V(T)) \subseteq L$, where L is the set of vertices already numbered by the eventual meo α. The next step in the iteration chooses a vertex $u \in V(T)$ of maximum cardinality. As in the MCS-M ordering [2], the cardinality is determined by the number of neighbors in the filled graph that have received their number in α. Subsequently, the iteration reorders the connected component $H = G(V(T))$ so that u is numbered last. Next, the iteration computes the new elimination subtree $T' = T'[u]$ obtained when the computed reordering is applied to H. The vertex u is the root of this new elimination subtree. Then u is ordered next by MCS-ETree and added to L. From the basic properties of elimination trees [20] and the fact that the loop invariant holds at the beginning of the iteration, we have the following for each elimination subtree $T'[c]$ added to $Trees$:

$$\begin{aligned}\mathrm{adj}_G(V(T'[c])) &\subseteq \{u\} \cup \mathrm{adj}_G(V(T'[u])) \\ &= \{u\} \cup \mathrm{adj}_G(V(T')) \\ &= \{u\} \cup \mathrm{adj}_G(V(T)) \\ &\subseteq L.\end{aligned}$$

The result then follows. ∎

At the beginning or end of any iteration, a *current ordering* γ is implicitly associated with the algorithm, as follows. For each vertex $x \in L$, we let $\gamma(x) = \alpha(x)$. The vertices in $V \setminus L$ are numbered from 1 to $n - |L|$ so that each child in an elimination subtree in $Trees$ receives a smaller number than that assigned to its parent. Then, at the beginning or end of any iteration, the *current filled graph* implicitly associated with the algorithm is G^+_γ.

Lemma 4.1 says that there are no edges in G joining two vertices from different elimination subtrees in $Trees$ at any point during the algorithm. It follows that the elimination subtrees generated throughout the algorithm will maintain their integrity, with no pair of elimination subtrees merged into a single elimination subtree by a current ordering γ associated with the algorithm. In other words, every elimination tree in $Trees$ is an elimination subtree of the elimination tree with respect to γ.

The key step within each iteration of the algorithm selects the vertex to receive the highest α-number among the vertices in the elimination subtree $T = T[v]$. For any vertex $x \in V(T)$, the set of vertices in L adjacent to x in the current filled graph G_γ^+ is $\text{hadj}_{G_\gamma^+}(x) \cap L$. What MCS-ETree requires is a maximum cardinality vertex in T with no descendants that are maximum cardinality vertices. That is, choose a vertex $u \in V(T)$ for which $|\text{hadj}_{G_\gamma^+}(u) \cap L| = |\text{hadj}_{G_\gamma^+}(v)|$ and for which $|\text{hadj}_{G_\gamma^+}(w) \cap L| < |\text{hadj}_{G_\gamma^+}(u) \cap L|$ for every descendant w of u. Since $\text{hadj}_{G_\gamma^+}(x) \cap L = \text{adj}_G(V(T[x])) \cap L$ for every vertex $x \in V(T)$, the algorithm equivalently chooses a vertex $u \in V(T)$ for which $|\text{adj}_G(V(T[u])) \cap L| = |\text{adj}_G(V(T))|$ and for which $|\text{adj}_G(V(T[w])) \cap L| < |\text{adj}_G(V(T[u])) \cap L|$ for every descendant w of u. Notice that such a vertex u always exists, since $\text{adj}_G(V(T[v])) \cap L = \text{adj}_G(V(T))$, and hence v can be chosen as u if no descendant x of v satisfies $|\text{adj}_G(V(T[x])) \cap L| = |\text{adj}_G(V(T))|$. Furthermore, if every descendant x of v satisfies $|\text{adj}_G(V(T[x])) \cap L| = |\text{adj}_G(V(T))|$, then u is chosen to be a leaf of T.

A second important step within each iteration uses algorithm Change_Root to compute a new peo of the filled graph of induced subgraph $H = G(V(T))$ under a topological ordering of T. This is instrumental in causing G_α^+ to be a subgraph of G_β^+.

LEMMA 4.2. *If γ is a current ordering for the algorithm at the beginning of an iteration and γ' is a current ordering for the algorithm at the end of the same iteration, then $E(G_{\gamma'}^+) \subseteq E(G_\gamma^+)$.*

Proof. Let $\{x, y\}$ be a fill edge in $E(G_{\gamma'}^+)$ at the end of the iteration. Let L be the set of numbered vertices at the beginning of the iteration. (Vertex u is added to L at the end of the iteration.) If both x and y belong to L, then by Lemma 2.1, we have $\{x, y\} \in E(G_\gamma^+)$. If either x or y is a vertex in one of the unnumbered elimination subtrees other than $T = T[v]$, then $\{x, y\} \in E(G_\gamma^+)$ because the subtree is ordered topologically by both γ and γ'. If both x and y are vertices in $V(T)$, then $\{x, y\} \in E(G_\gamma^+)$ because $H = G(V(T))$ is renumbered by algorithm Change_Root with a peo of the filled graph $H_{\gamma_1}^+$, where γ_1 is a topological ordering of T. Finally, the only case remaining is where $x \in V(T)$ and $y \in L$. If $x \notin \text{anc}_T[u]$, then from algorithm Change_Root, we have $T'[x] = T[x]$, so by Lemma 2.1, we have $\{x, y\} \in E(G_\gamma^+)$. If $x \in \text{anc}_T[u]$, then by the choice of u in the algorithm and the simple properties of elimination trees [20],

$$\begin{aligned}
\text{adj}_{G_{\gamma'}^+}(x) \cap L &= \text{adj}_G(V(T'[x])) \cap L \\
&\subseteq \text{adj}_G(V(T')) \\
&= \text{adj}_G(V(T)) \\
&= \text{adj}_G(V(T[u])) \cap L \\
&\subseteq \text{adj}_G(V(T[x])) \cap L \\
&= \text{adj}_{G_\gamma^+}(x) \cap L.
\end{aligned}$$

This completes the proof. □

Observe that Lemma 4.2 holds for the sequence of filled graphs associated with the sequence of current orderings in Figure 4.2. Observe also that in the final filled graph in Figure 4.2, fill edge $\{4, 5\}$ is the sole chord of a 4-cycle joining vertices 3, 5, 6, and 4, and fill edge $\{2, 5\}$ is the sole chord of a 4-cycle joining vertices 1, 5, 3, and 2. In our proof of correctness, we will show that every fill edge in the final filled graph G_α^+ generated by MCS-ETree is the sole chord of a 4-cycle and then use Theorem 2.2 to argue the result.

THEOREM 4.3. *Given a graph G and an ordering β of G, algorithm MCS-ETree generates an meo α of G such that G^+_α is a subgraph of G^+_β.*

Proof. From Lemma 4.2 and the definition of current orderings γ within the algorithm, it follows that G^+_α is a subgraph of G^+_β. If there are no fill edges in G^+_α, then G is chordal and α is a peo of G. It follows that α is an meo of G, and hence we have the result in this case. Assume therefore that G^+_α has at least one fill edge. Let $\{u, w\}$ be a fill edge in G^+_α. We will find a 4-cycle in G^+_α for which $\{u, w\}$ is the only chord. The minimality of the ordering α will then follow by Theorem 2.2.

Without loss of generality, assume that $\alpha(u) < \alpha(w)$. Let $T = T[v]$ be the elimination subtree that the algorithm is processing when u is chosen by the algorithm to receive its α-number. Let γ be a current ordering for the algorithm at the beginning of this iteration. By Lemma 4.2, since $\{u, w\}$ is a fill edge in G^+_α, $\{u, w\}$ is also a fill edge in the current filled graph G^+_γ. By Lemma 2.1, there is a fill path $u - x_1 - \cdots - x_r - w$ ($r \geq 1$) in G through vertices x_i that are the descendants of u in T. Notice that u therefore cannot be a leaf of T. The vertices x_i come from one (and only one) of the subtrees rooted at a child of u in T, say, $T[c]$. Let x_t be the vertex among the x_i that is eventually numbered highest by the algorithm. Then there are fill paths (or direct edges) in G from x_t to u and from x_t to w under the final ordering α generated by the algorithm. So $\{x_t, u\}$ and $\{x_t, w\}$ are edges in the final filled graph G^+_α.

By Lemma 2.1, we know that $\text{hadj}_{G^+_\gamma}(c) \cap L \subseteq \text{hadj}_{G^+_\gamma}(u) \cap L$ for each child c of u in T. From the choice of u, we can conclude that $\text{hadj}_{G^+_\gamma}(c) \cap L \subset \text{hadj}_{G^+_\gamma}(u) \cap L$ (proper subset). Let $y \in (\text{hadj}_{G^+_\gamma}(u) \cap L) \setminus (\text{hadj}_{G^+_\gamma}(c) \cap L)$, which is not empty. By Lemma 2.1, none of the vertices x_i is adjacent to y in G^+_γ (including x_t). By Lemma 4.2, $\{x_t, y\}$ is not an edge in the final filled graph G^+_α.

Finally, the set $\text{hadj}_{G^+_\gamma}(u) \cap L = \text{hadj}_{G^+_\gamma}(v)$ is the higher adjacency set of u in the final filled graph G^+_α, since u receives its number at this step, and all vertices that are numbered higher than u in α have already received their numbers. So $\text{hadj}_{G^+_\gamma}(u) \cap L = \text{hadj}_{G^+_\alpha}(u)$, and recall that $\text{hadj}_{G^+_\alpha}(u) \cup \{u\}$ is a clique in G^+_α. Note that both y and w belong to $\text{hadj}_{G^+_\alpha}(u)$. It follows that $\{u, y\}$ and $\{w, y\}$ are both edges in G^+_α. This completes a 4-cycle $x_t - u - y - w - x_t$ in G^+_α for which $\{u, w\}$ is the only chord.

Since every fill edge is the only chord of such a 4-cycle, the final filled graph G^+_α is a minimal chordal supergraph by Theorem 2.2, and the final ordering, which is a peo of G^+_α, is an meo of G. ∎

5. Implementation details and running time analysis. We can adapt basic tools from sparse matrix computations to obtain a time bound of $O(nm\,\text{A}(m, n))$ for algorithm MCS-ETree.

THEOREM 5.1. *The running time of algorithm MCS-ETree is $O(nm\,A(m, n))$.*

Proof. Let us consider the three major tasks the algorithm must perform as it goes through a single iteration of the *while* loop. Let γ be a current ordering at the beginning of the iteration, and let $T = T[v]$ be the elimination subtree chosen to be processed.

First, the algorithm needs the values $|\text{hadj}_{G^+_\gamma}(x) \cap L| = |\text{adj}_G(V(T[x])) \cap L|$ for every vertex $x \in V(T)$. One option is to compute and work directly with the filled graph G^+_γ, but this leads to $O(nm')$ total work for this task (summed over all iterations of the *while* loop), where m' is the number of edges in the initial filled graph G^+_β. It also requires the storage of filled graphs rather than just the original graph G. We have not implemented this option. Gilbert, Ng, and Peyton [12] introduced a fast algorithm for computing the number of nonzeros in each row and each column of a sparse Cholesky factor. Hence a second, and better, option is to modify the

algorithm in [12] for column nonzero counts to compute the values $|\text{hadj}_{G_\gamma^+}(x) \cap L|$ for every vertex $x \in V(T)$. This option leads to $O(nm\ \text{A}(m,n))$ total work for this task, summed over all iterations of the *while* loop. It does not involve or require the computation of any filled graphs explicitly.

The algorithm in [12] is geared to compute $|\text{ladj}_{G_\gamma^+}(x) \cup \{x\}|$ (the "row count") and $|\text{hadj}_{G_\gamma^+}(x) \cup \{x\}|$ (the "column count") for every vertex $x \in V(G)$. In adapting for use by MCS-ETree, the computation is restricted in three different ways. First, none of the computation connected with row counts is carried out. Second, the computation can be restricted to the elimination subtree $T = T[v]$ processed by the current iteration of the algorithm rather than the entire elimination tree associated with a current ordering γ. And third, the counts must be restricted to compute $|\text{hadj}_{G_\gamma^+}(x) \cap L|$ rather than $|\text{hadj}_{G_\gamma^+}(x) \cup \{x\}|$. It is straightforward to adapt the implementation in Gilbert, Ng, and Peyton [12, page 1085] to incorporate these restrictions. Note also that a postordering of the elimination subtree T is required by our adaptation of the algorithm. This requirement is inherited from the original algorithm.

Second, MCS-ETree uses algorithm Change_Root to reorder the vertices of T so that u becomes the new root and there is no additional fill under the new current ordering. To implement Change_Root, we initially reorder the vertices of T by a postordering that numbers each vertex in $\text{anc}_T[u]$ before any of its siblings. The ordering and marking process can then be performed as the vertices are visited in this postorder. The total work spent on this task over all iterations of the outer loop is $O(nm)$.

Third, the algorithm needs to recompute the elimination subtree for the new ordering of $H = G(V(T))$. The elimination subtree can be computed with a single sweep of the full adjacency lists of the vertices of T and the required disjoint set union operations. It is trivial to adapt the standard algorithm [20] for computing the entire elimination tree to compute the elimination subtrees needed here. The total work for this task over all iterations of the outer loop is $O(nm\ \text{A}(m,n))$. This concludes the proof. □

5.1. Basic implementation. We have implemented these three steps in the most straightforward way possible, with no attempt at avoiding redundant work. The object with the first implementation was to make it as simple as possible. We have called this first implementation the *basic* implementation.

With the basic implementation established, we sought to enhance the implementation by avoiding redundant work. There is much redundant work to be avoided in all three of the major steps within each iteration. Getting rid of this redundant work does not reduce the overall provable time bound of the algorithm, but it results in a much faster implementation in practice, as the test results will show in the next section.

5.2. Enhanced implementation. Consider again the computation of the values $|\text{hadj}_{G_\gamma^+}(x) \cap L|$ for every vertex $x \in V(T[v])$. A key part of the algorithm in [12] is the recognition of and reduction to the so-called *skeleton adjacency sets* [18] associated with the current ordering. If these sets are known and stored ahead of the computation, then they can be traversed rather than using full adjacency sets. Let $z \in L$, and let $T[v]$ be the current elimination subtree. Let $T_r[z,v]$ denote the *row subtree* of z in $T[v]$. That is, $V(T_r[z,v])$ is the set of the vertices of $T[v]$ that are adjacent to z in the current filled graph G_γ^+. We say that z is in the *skeleton adjacency set* of $x \in V(T[v])$ if x is a leaf of $T_r[z,v]$. Note that our skeleton adjacency set of $x \in V(T[v])$ is limited

to vertices in L. To ultimately improve efficiency, we store the skeleton adjacency sets of all vertices $x \in V(T[v])$ as the values $|\text{hadj}_{G^+_\gamma}(x) \cap L|$ are computed. The skeleton adjacency sets come as a natural by-product of the computation.

In Figure 4.2, consider the point where the vertex of maximum cardinality is chosen from the unnumbered elimination subtree $T'[c]$, whose root has current γ-number 4. The skeleton adjacency sets of vertices 1, 2, 3, and 4 are $\{5\}$, $\{5\}$, $\emptyset$, and $\{6\}$, respectively. These skeleton adjacency sets suffice to compute the needed cardinalities.

Again, let $T = T[v]$. For each vertex $x \in \text{anc}_T[u]$, it is possible that $T'[x] \neq T[x]$ because of the reordering obtained from algorithm Change_Root (if $u \neq v$). So the vertices in the skeleton adjacency set of x cannot safely be used during the next step that processes the subtree containing x. The entire adjacency set of x must be used during the next step that processes the subtree containing x. But in the case where $x \in V(T) \setminus \text{anc}_T[u]$, we have $T'[x] = T[x]$ because of the reordering obtained from algorithm Change_Root. The descendants of x remain precisely the same, so the skeleton adjacency set of x does not change, except for the possible addition of the new root u. We take care of the update with u and process the old abbreviated skeleton adjacency set during the next step that processes the subtree containing x.

So in summary, we process abbreviated skeleton adjacency sets, many of which are in practice empty, for vertices that at the most recent relevant step were in $V(T) \setminus \text{anc}_T[u]$; we process full adjacency sets for vertices that at the most recent relevant step were in $\text{anc}_T[u]$. To store the skeleton adjacency sets requires another vector large enough to store the full adjacency structure of G. But this technique promises to improve run times appreciably.

Consider again how to implement algorithm Change_Root for computing a reordering of an elimination tree T so that $u \in V(T)$ becomes the new root and no new fill is introduced. As before, we reorder the vertices of T with a postordering for which every vertex in $\text{anc}_T[u]$ is numbered before any of its siblings. The procedure Change_Root2 in Figure 5.1 can then be used to perform the reordering. The prescribed postordering is input as γ_1, which is, of course, a topological ordering of T. Unlike our earlier implementation of algorithm Change_Root, the only adjacency sets that algorithm Change_Root2 traverses are those for vertices in $\text{anc}_T[u]$. This also promises to improve run times appreciably.

Consider the recomputation of the elimination subtree, replacing $T = T[v]$ with $T' = T'[u]$. Again, the subtrees rooted at vertices in $V(T) \setminus \text{anc}_T[u]$ remain unchanged as MCS-ETree goes forward to the next iteration. So there is no need to recompute these portions of the elimination subtree. The new subtree can be patched together with an enhanced implementation that traverses the adjacency sets of the vertices of $\text{anc}_T[u]$ only.

These enhancements do not change the time complexity of the algorithm; it remains $O(nm\,\text{A}(m,n))$. We call the improved implementation of the algorithm the *enhanced* implementation. Because the components of the work of lower time complexity have greater relative influence on performance after these enhancements are incorporated, there are other improvements implemented in marking processes and initializations. These are not described here.

5.3. Blocked implementation. Finally, there is one further enhancement of a completely different sort to incorporate into the code. For this last version, we first include all of the enhancements described thus far, then we add the following. When $T = T[v]$ is processed and vertex u is to be numbered, we can often detect other

```
procedure Change_Root2(G, T, γ1, u, γ2)
Input: A graph G, an elimination tree T of G with respect to γ1,
       a prescribed postordering γ1 of T, and a vertex u ∈ V(T).
Output: An equivalent reordering γ2 of G with respect to γ1,
        where u is numbered last.
  for i ∈ [0, 1, ..., |V(T)|]; B(i) ← ∅; end for;
  j ← 0;
  for x ∈ V(T) in the prescribed postorder γ1
    if x ∈ V(T) \ anc_T[u] then
      j ← j + 1; γ2(x) ← j;
    end if;
  end for;
  B(0) ← B(0) ∪ {u};
  for x ∈ anc_T(u)
    j ← |V(T)|;
    for y ∈ adj_G(x)
      j ← min(j, γ1(y));
    end for;
    B(j) ← B(j) ∪ {x};
  end for;
  j ← |V(T)|;
  for i ∈ [0, 1, ..., |V(T)|] in order
    for x ∈ B(i)
      γ2(x) ← j;
      j ← j − 1;
    end for;
  end for;
end Change_Root2;
```

FIG. 5.1. *An enhanced variant of algorithm Change_Root for the second step in the main loop of MCS-ETree.*

vertices among the vertices of $\mathrm{anc}_T(u)$ that can be numbered in a block along with u and removed with no further processing. There are two cases to consider. First, consider the case where u has descendants in T. Let $c_1, \ldots, c_r$ be the children of u in T. If a vertex $x \in \mathrm{anc}_T(u)$ is adjacent to each subtree $T[c_1], \ldots, T[c_r]$ and the adjacency set of x contains every vertex that is in the skeleton adjacency set of u (again, limited to skeleton neighbors in L), then x can be ordered in a block along with u (see Lemma 5.2). Having possession of the skeleton adjacency sets is crucial here for implementing detection of this condition. These are available only after our enhancement for the computation of cardinalities.

In Figure 4.2, consider the unnumbered elimination subtree $T'[c]$, whose root has γ-number 5. There, the maximum cardinality vertex chosen is vertex 4. The sole member of the skeleton adjacency set of vertex 4 is vertex 6. Since vertex 5 is adjacent in G to vertex 6 and also adjacent in G to the subtree rooted at vertex 3, it follows that vertex 5 can be ordered in a block along with vertex 4.

Second, consider the case where u has no descendants in T. If a vertex $x \in \mathrm{anc}_T(u)$ is adjacent to u and every vertex in $\mathrm{adj}_G(u)$ (except x, of course), then x can be ordered in a block along with u (see Lemma 5.3).

LEMMA 5.2. *Let u be a vertex of maximum cardinality chosen at some iteration of algorithm MCS-ETree such that u has descendants in $T = T[v]$. Let X be the set*

comprised of u and any vertex $x \in anc_T(u)$ adjacent to all of the subtrees rooted at children of u and adjacent to all of the members of u's skeleton adjacency set. Our algorithm can be modified so that it numbers next as a block the vertices in X in the current iteration.

Proof. Let the algorithm be modified so that it numbers the vertices of X next as a block in the current iteration. Let L be the set of numbered vertices before the vertices of X are numbered. Note first that by the choice of u, the definition of X, and Lemma 2.1, every vertex of X will be adjacent to every vertex of $\mathrm{adj}_G(V(T))$ in the final filled graph. Choose $x \in X$, and let $\{x, w\}$ be a fill edge, where w is numbered higher than x by the ordering. (Note that x may be u.) For our first case, suppose that $w \in L$. Note that w is not in u's skeleton adjacency set, otherwise w would be in x's adjacency set, and hence we would not have a fill edge. This means that w is adjacent to one of the subtrees rooted at a child c of u. Since x is adjacent to every vertex of $\mathrm{adj}_G(V(T))$ in the final fill graph and x is also adjacent to $T[c]$, this means that we can argue, just as in the proof of correctness, the existence of a 4-cycle that has the fill edge $\{x, w\}$ as its sole chord.

For our second case, suppose that $w \in X$. Since both x and w are adjacent to all subtrees rooted at children of u, we can again argue, as above and in the proof of correctness, the existence of a 4-cycle that has the fill edge $\{x, w\}$ as its sole chord. □

LEMMA 5.3. *Let u be a vertex of maximum cardinality chosen at some iteration of MCS-ETree such that u has no descendants in $T = T[v]$. Any vertex $x \in anc_T(u)$ that is adjacent to u and every vertex in $adj_G(u)$ (except x) can be ordered in a block along with u.*

Proof. In this case, there is no fill edge incident to x and a higher-numbered vertex, so the result follows. □

Based on Lemma 5.2, we modified MCS-ETree to number all vertices of any block X described by the lemma at the end of the current iteration. Based on Lemma 5.3, we also modified MCS-ETree to number all vertices of any block described by the lemma at the end of the current iteration. We call our implementation that includes all of the previous enhancements and this capability to number blocks of vertices the *blocked* implementation. The detection of the blocks is implemented by additional code within the Change_Root2 procedure that does not require any further traversal of adjacency sets. The vertices of a block are placed in the set $B(0)$, where they are labeled last by ordering γ_2 among the vertices of the current elimination subtree.

6. Test results. We have coded the *basic*, *enhanced*, and *blocked* implementations discussed in section 5. For test results in an earlier technical report [15], we ran these implementations on a set of test problems taken from the Harwell–Boeing collection of sparse matrices. The initial orderings used in [15] were approximate minimum degree (AMD) [1] orderings and random orderings. As reported in earlier work [5, 24], minimum degree orderings are so close to minimal in practice that there is very little extraneous fill to remove. Consequently, the practical impact of MCS-ETree is extremely limited when AMD initial orderings are used. But our timing results in [15] indicate that the best implementation of MCS-ETree is very efficient on AMD initial orderings. A full set of tables and a discussion of results on AMD and random initial orderings can be found in our technical report [15].

In this paper, we run our three implementations of MCS-ETree on a set of test problems taken from the sparse matrix collection of Tim Davis. A greater variety of stuctural analysis problems are included, along with a number of problems from optimization and other application areas. The structural analysis problems include

TABLE 6.1
The number of vertices in each graph, the number of edges in each filled graph, and the number of factorization operations when the initial ordering is ND.

Matrix	$\|V\|$	Edges in filled graph			Factorization operations		
		ND ($\times 10^3$)	MCS-ETree ($\times 10^3$)	% decr.	ND ($\times 10^6$)	MCS-ETree ($\times 10^6$)	% decr.
BCSSTK17	10974	1126	1061	5.81	191.2	162.6	14.95
BCSSTK25	15439	1541	1434	6.89	350.4	286.1	18.35
SRBEDDY	46772	7527	7057	6.25	2190.1	1764.6	19.43
CRYSTK02	13965	4248	4205	1.02	1923.5	1863.2	3.13
CRYSTK03	24696	9508	9390	1.24	5631.4	5388.9	4.31
nasasrb	54870	10505	10011	4.70	3559.9	3055.4	14.17
pkustk01	22044	2075	2053	1.04	421.9	414.9	1.66
pwt	36519	1346	1341	0.39	110.9	110.3	0.52
shuttle_eddy	10429	352	329	6.58	22.2	17.9	19.00
skirt	12598	466	453	2.65	31.1	29.5	5.08
tandem_dual	94069	6481	6466	0.23	2265.0	2260.7	0.19
helm3d01	32226	4914	4900	0.29	2783.9	2773.5	0.37
pli	22695	13842	13799	0.31	15845.2	15813.4	0.20
Pres_Poisson	14822	2387	2338	2.05	553.2	522.1	5.63
ex3sta1	16782	7748	7658	1.16	7440.5	7276.8	2.20
fxm4_6	18892	435	424	2.65	23.5	22.5	4.53
gupta1	31802	2014	1987	1.33	297.1	270.9	8.83
minsurfo	40806	954	952	0.23	97.5	97.3	0.22
nemeth02	9506	460	220	52.05	24.5	5.8	76.35
pfinan512	74752	1748	1723	1.45	163.0	156.4	4.08
ted_B	10605	87	72	17.05	1.5	1.1	29.23
vibrobox	12328	2483	2466	0.68	1318.3	1310.4	0.60

BCSSTK17, BCSSTK25, SRBEDDY, CRYSTK02, CRYSTK03, nasasrb, pkustk01, pwt, shuttle_eddy, skirt, tandem_dual, and pli. The optimization problems include ex3sta1, fxm4_6, gupta1, minsurfo, and pfinan512. Also included are Helmholtz equations on a unit cube (helm3d01), a CFD problem (Pres_Poisson), a quantum chemistry problem (nemeth02), a thermoelasticity problem (ted_B), and an acoustics problem (vibrobox).

Also, we look at nested dissection (ND) initial orderings only. For ND initial orderings, there is sometimes a significant amount of extraneous fill to remove; hence MCS-ETree is tested in a more demanding setting and gives results of more practical consequence. We also run a code that implements the algorithm from Peyton [24] for solving the same problem and compare our algorithm with this algorithm, since it has the fastest documented practical running time. We remind the reader that the theoretical running time bound of the algorithm of [24] is not known.

Table 6.1 reports the number of vertices in each graph and the number of edges in the filled graphs for the ND orderings and the minimal orderings obtained from the *blocked* implementation of MCS-ETree. Also shown are the number of factorization operations that result from the ND orderings and the minimal orderings. The percent decrease in edges is less than 2% for 12 of the 22 problems. It is greater than 5% for 6 of the 22 problems. For matrix ted_B, which comes from coupled linear thermoelasticity equations, the decrease is 17%. For nemeth02, which comes from a Newton–Schultz iteration for a chemistry problem, the decrease is 52%. The latter is a long "path-like" problem for which nested dissection is inappropriate unless one seeks to exploit parallelism during the factorization.

Looking at the reductions in factorization operations gives us more matrices where MCS-ETree has some practical impact. Nonetheless, the reduction is less than 5%

TABLE 6.2

CPU seconds to compute the ND initial orderings and the minimal orderings using the algorithm of [24] *and the three implementations of MCS-ETree.*

Matrix	ND time	MCS-ETree (*Basic*) time	MCS-ETree (*Enhanced*) time	MCS-ETree (*Blocked*) time	Peyton [24] time
BCSSTK17	0.124	9.769	0.820	0.052	0.496
BCSSTK25	0.256	10.141	1.688	0.088	0.604
SRBEDDY	0.188	129.492	10.225	0.280	0.648
CRYSTK02	0.144	38.538	3.016	0.088	0.144
CRYSTK03	0.272	98.350	7.800	0.152	0.276
nasasrb	1.056	225.106	16.173	0.368	4.044
pkustk01	0.076	24.914	2.332	0.156	1.316
pwt	0.420	12.029	2.160	0.144	0.304
shuttle_eddy	0.096	2.356	0.400	0.040	0.088
skirt	0.148	1.952	0.316	0.052	0.556
tandem_dual	1.236	72.601	22.221	0.472	2.772
helm3d01	0.580	62.580	12.829	0.324	1.988
pli	0.972	94.538	11.277	0.148	1.776
Pres_Poisson	0.148	22.333	1.680	0.080	0.184
ex3sta1	0.336	64.776	10.205	0.240	3.056
fxm4_6	0.248	5.064	0.660	0.104	0.768
gupta1	1.680	38.178	17.565	6.876	17148.713
minsurfo	0.355	12.854	3.172	0.188	0.369
nemeth02	0.219	44.676	2.082	1.859	2.553
pfinan512	1.188	20.822	3.971	0.324	4.281
ted_B	0.057	2.057	0.168	0.076	0.148
vibrobox	0.293	17.010	2.592	0.080	0.867

for 12 of the 22 problems. For three of the structural analysis matrices used also in our technical report [15], namely, BCSSTK17, BCSSTK25, and SRBEDDY, the decreases are roughly 15%, 18%, and 19.5%, respectively. Two structural analysis matrices added to our problem set for this paper, namely, nasasrb and shuttle_eddy, have decreases of roughly 14% and 19%, respectively. The two matrices with the largest fill reductions, namely, ted_B and nemeth02, have decreases of roughly 29% and 76%, respectively.

Table 6.2 reports the CPU time in seconds for the ND orderings, for each of the three implementations of algorithm MCS-ETree, and for the algorithm of [24]. The tests were run on a PC with a Pentium 4 processor running at 2.66 GHz with 0.99 GB RAM available. The code was written in Fortran and executed under the Linux operating system. We used the Metis software package available from the University of Minnesota to compute nested dissection orderings.

The basic implementation of MCS-ETree has much larger run times than the ND code and is clearly too inefficient for practical sparse matrix computations. The enhanced implementation of MCS-ETree is much faster than the basic implementation in every case. Often it is ten times faster, or close to ten times faster, than the basic implementation. But comparing run times for the enhanced implementation with the ND ordering times, it is obvious that the enhanced implementation is also too inefficient for practical sparse matrix computations, despite its improvements.

Our timings, however, improve dramatically for most problems as we move to the blocked implementation, which includes the blocking technique along with all of the enhancements employed by the enhanced implementation. For 17 of the 22 problems, the blocked implementation runs more than ten times faster than the enhanced im-

plementation. For 13 of the 22 problems, the blocked implementation runs more than 100 times faster than the basic implemenation.

The three problems for which the reduction in time from the basic implementation to the blocked implementation is smallest are gupta1, nemeth02, and ted_B. For each of these problems, the number of blocks detected by the blocked implementation is unusually large relative to the number of vertices in the graph. For gupta1, there are 31,198 blocks and 31,802 vertices; for nemeth02, there are 7,619 blocks and 9506 vertices; for ted_B, there are 8,750 blocks and 10,605 vertices. For nemeth02, the reduction from the enhanced implementation to the blocked implementation is very small—from 2.082 seconds to 1.859 seconds. The matrix gupta1, which arises in a linear programming problem, presents the greatest difficulties to all of the algorithms we have looked at. For gupta1, the reduction from the basic implementation to the blocked implementation is from 38.178 seconds to 6.876 seconds—a reduction of only 82%. The time for the ND ordering of gupta1 is also greater than the ND ordering time for any other matrix. The matrices nemeth02 and ted_B have, by far, the greatest percent reduction in edges, and it may be natural to pay more in time to remove a greater percentage of edges.

The algorithm of Peyton [24] runs reasonably fast for ND initial orderings (except on the matrix gupta1). It is not as fast as the blocked implementation of MCS-ETree for any test matrix when ND initial orderings are used; however, there are a few instances where it is faster when AMD initial orderings are used [15]. In fairness, the implementation of the algorithm of [24] has not been improved to the extent that the blocked implementation of MCS-ETree has been improved. It would be interesting to see if the implementation of the algorithm of [24] could be improved to the extent that it would prove more competitive than the blocked implementation of MCS-ETree for ND initial orderings.

The algorithm of [24] takes an exhorbitant amount of time (4.76 hours) to compute the minimal ordering for the matrix gupta1. It became clear to us what was going on when we saw that the AMD ordering time for gupta1 is 52.2 seconds, which is extremely large. It was shown in Heggernes et al. [14] that AMD is an $O(nm)$ algorithm and that there exist examples to which this worst-case time complexity applies. The algorithm of [24] relies on the iteration of a restricted version of the minimum degree algorithm with exact degrees. The time complexity of minimum degree with exact degrees is $O(n^2m)$ [14]. This interesting test matrix fully reveals the vulnerability of the algorithm of [24].

Finally, run times for the blocked implementation are reduced to the point that MCS-ETree is fast enough to be considered for sparse matrix computations. For 16 of 22 problems, the time required by the blocked implementation of MCS-ETree is less than that required by the ND algorithm. For only two of the 22 problems, the ratio of the time for the blocked implementation to the time for the ND ordering is greater than three; for gupta1 the ratio is 4.09, and for nemeth02, the ratio is 8.50.

7. Concluding remarks. We have introduced a new algorithm MCS-ETree for computing a minimal ordering whose minimal fill lies inside the fill of any given initial ordering. The $O(nm\,\mathrm{A}(m,n))$ running time complexity is virtually as good as the best known time complexity of $O(nm)$. In practical tests, our algorithm performs better than the previous fastest algorithm of [24] and has the advantage of having a provably good theoretical running time as well. Algorithm MCS-ETree explicitly deals with a current ordering and the structure associated with that ordering, at the cost of disjoint set union operations that lead to the extremely slowly growing $\mathrm{A}(m,n)$ term

in its running time complexity. By explicitly computing and exploiting elimination subtrees and partial Cholesky column nonzero counts, one obtains a relatively simple algorithm whose proof of correctness is also relatively simple. The new algorithm is based on selecting a special vertex of maximum cardinality at each step and resembles, in this regard, the algorithm MCS-M introduced in [2].

The algorithm can be implemented in $O(nm\ \mathrm{A}(m,n))$ time by adapting three commonly used sparse matrix algorithms that date from the mid-1980's to the mid-1990's:

1. An $O(m\ \mathrm{A}(m,n))$ algorithm for computing the number of nonzeros in each column of a Cholesky factor [12];
2. An $O(m)$ algorithm for computing equivalent reorderings [19]; and
3. An $O(m\ \mathrm{A}(m,n))$ algorithm for computing an elimination tree [20].

We were able to improve the *basic* implementation to obtain much faster implementations. The first set of enhancements are straightforward programming-level improvements that greatly limit the number of times adjacency lists are traversed or shorten those lists to abbreviated skeleton adjacency lists. The other improvement allows blocks of vertices to be numbered by a single iteration of the algorithm, and this is based closely on the idea of indistinguishable vertex sets in elimination graphs exploited so successfully by the implementations of the minimum degree algorithm [17].

We coded in Fortran the *basic*, *enhanced*, and *blocked* implementations, and our timing results show that the blocked implementation is fast enough to be considered for use in sparse matrix computations. The best implementation of the algorithm could prove useful when the initial orderings are nested dissection orderings, because sometimes the fill and factorization operations can be significantly reduced by removing extraneous fill.

Acknowledgments. The authors thank the referees for their many helpful suggestions. We also thank Vince Postell for his help in gaining access to Linux and PSTricks.

REFERENCES

[1] P. R. Amestoy, T. A. Davis, and I. S. Duff, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905.

[2] A. Berry, J. R. S. Blair, P. Heggernes, and B. W. Peyton, *Maximum cardinality search for computing minimal triangulations of graphs*, Algorithmica, 39 (2004), pp. 287–298.

[3] A. Berry, J.-P. Bordat, P. Heggernes, G. Simonet, and Y. Villanger, *A wide-range algorithm for minimal triangulation from an arbitrary ordering*, J. Algorithms, 58 (2006), pp. 33–66.

[4] A. Berry, P. Heggernes, and G. Simonet, *The minimum degree heuristic and the minimal triangulation process*, in Graph-theoretic Concepts in Computer Science, 29th International Workshop, Lect. Notes Comput. Sci. 2880, H. Bodlaender, ed., Springer, New York, 2003, pp. 58–70.

[5] J. R. S. Blair, P. Heggernes, and J. A. Telle, *A practical algorithm for making filled graphs minimal*, Theoret. Comput. Sci., 250 (2001), pp. 125–141.

[6] H. L. Bodlaender and A. M. C. A. Koster, *Safe separators for treewidth*, Discrete Math., 306 (2006), pp. 337–350.

[7] F. R. K. Chung and D. Mumford, *Chordal completions of planar graphs*, J. Combin. Theory Ser. B, 31 (1994), pp. 96–106.

[8] E. Dahlhaus, *Minimal elimination ordering inside a given chordal graph*, in Graph-theoretical Concepts in Computer Science, 23rd International Workshop, Lect. Notes Comput. Sci. 1335, R. Möhring, ed., Springer, New York, 1997, pp. 132–143.

[9] F. V. Fomin, D. Kratsch, and I. Todinca, *Exact (exponential) algorithms for treewidth and minimum fill-in*, in Automata, Languages and Programming, Lect. Notes Comput. Sci. 3142, J. Diaz, J. Karhumäki, A. Lepisto, and D. Sanella, eds., ICALP 2004, Springer, New York, 2004, pp. 568–580.

[10] D. Fulkerson and O. Gross, *Incidence matrices and interval graphs*, Pacific J. Math., 15 (1965), pp. 835–855.
[11] A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363.
[12] J. R. Gilbert, E. G. Ng, and B. W. Peyton, *An efficient algorithm to compute row and column counts for sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1075–1091.
[13] P. Heggernes, *Minimal triangulations of graphs: A survey*, Discrete Math., 306 (2006), pp. 297–317.
[14] P. Heggernes, S. Eisenstat, G. Kumfert, and A. Pothen, *The computational complexity of the Minimum Degree algorithm*, in Proceedings of the 14th Norwegian Computer Science Conference, NIK 2001, University of Tromso, Norway, 2001.
[15] P. Heggerenes and B. W. Peyton, *Fast computation of minimal fill inside a given elimination ordering*, Technical report Inform. 343, University of Bergen, 2007.
[16] S. L. Lauritzen and D. J. Spiegelhalter, *Local computations with probabilities on graphical structures and their applications to expert systems*, J. Roy. Statist. Soc. Ser. B, 50 (1988), pp. 157–224.
[17] J. W. H. Liu, *Modification of the minimum degree algorithm by multiple elimination*, ACM Trans. Math. Software, 11 (1985), pp. 141–153.
[18] J. W. H. Liu, *A compact row storage scheme for Cholesky factors using elimination trees*, ACM Trans. Math. Software, 12 (1986), pp. 127–148.
[19] J. W. H. Liu, *Equivalent sparse matrix reorderings by elimination tree rotations*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 424–444.
[20] J. W. H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.
[21] A. Natanzon, R. Shamir, and R. Sharan, *A polynomial approximation algorithm for the minimum fill-in problem*, SIAM J. Comput., 30 (2000), pp. 1067–1079.
[22] T. Nedretvedt, *Implementation of a Minimal Triangulation Algorithm for Studying Properties of the Minimum Degree Heuristic*, Master's thesis, University of Bergen, Norway, 2005.
[23] S. Parter, *The use of linear graphs in Gauss elimination*, SIAM Rev., 3 (1961), pp. 119–130.
[24] B. W. Peyton, *Minimal orderings revisited*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 271–294.
[25] D. J. Rose, R. E. Tarjan, and G. S. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.
[26] D. J. Rose, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, in Graph Theory and Computing, R. C. Read, ed., Academic Press, New York, 1972, pp. 183–217.
[27] R. E. Tarjan and M. Yannakakis, *Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM J. Comput., 13 (1984), pp. 566–579.
[28] M. Yannakakis, *Computing the minimum fill-in is NP-complete*, SIAM J. Algebraic Discrete Methods, 2 (1981), pp. 77–79.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1445–1462

A FAMILY OF RATIONAL ITERATIONS AND ITS APPLICATION TO THE COMPUTATION OF THE MATRIX pTH ROOT*

BRUNO IANNAZZO†

Abstract. Matrix fixed-point iterations $z_{n+1} = \psi(z_n)$ defined by a rational function ψ are considered. For these iterations a new proof is given that matrix convergence is essentially reduced to scalar convergence. It is shown that the principal Padé family of iterations for the matrix sign function and the matrix square root is a special case of a family of rational iterations due to Ernst Schröder. This characterization provides a family of iterations for the matrix pth root which preserve the structure of a group of automorphisms associated with a bilinear or a sesquilinear form. The first iteration in that family is the Halley method for which a convergence result is proved. Finally, new algorithms for the matrix pth root based on the Newton and Halley iterations are designed using the idea of the Schur–Newton method of Guo and Higham.

Key words. Halley's method, matrix iteration, matrix root, matrix function, Newton's method, rational iterations, structure-preserving

AMS subject classifications. 65F30, 15A15

DOI. 10.1137/070694351

1. Introduction.

The study of rational iterations, which have the form $x_{k+1} = \varphi(x_k)$, where $\varphi(z)$ is a rational function, is a topic of great interest in computation, particularly for the design and analysis of root-finding algorithms. The local convergence at a fixed point z_*, such that $z_* = \varphi(z_*)$, is related to the properties of the derivatives of φ at z_*. A study of the global convergence is very difficult: the sets of initial values for which the sequence generated by a rational iteration converges to a fixed point are bounded by the so-called Julia sets which in most cases are fractals [1].

The generalization to the matrix case appears in the study of matrix equations and in the computation of matrix functions [9]. It raises problems somehow new: it is not straightforward how to define a rational matrix iteration; there can be infinite fixed points, the lack of commutativity in finite arithmetic can have effects on the convergence, and so on.

In this paper we provide a general convergence result for rational matrix iterations, and then we prove some properties of specific classes of rational iterations.

General results concern the case where the iterates are rational functions of a matrix A, say, $s_k(A)$. We prove that the uniform convergence of $s_k(z)$ on a compact neighborhood of the spectrum of A implies the matrix convergence. Then we show that if the iteration is of the type $x_{k+1} = \varphi(x_k)$, where φ is a rational function, then the pointwise convergence of $s_k(\lambda)$ to attractive fixed points for each eigenvalue λ of A implies the uniform convergence on a compact neighborhood of the spectrum of A and thus the matrix convergence. This extends in part a result of Higham [9, Thm. 4.15].

*Received by the editors June 12, 2007; accepted for publication (in revised form) by V. Simoncini July 21, 2008; published electronically December 3, 2008. This work was partially supported by MIUR grant 2006017542.

http://www.siam.org/journals/simax/30-4/69435.html

†Dipartimento di Fisica e Matematica, Università dell'Insubria, Via Valleggio 11, 22100 Como, Italy (bruno.iannazzo@uninsubria.it).

Concerning specific classes, we first consider the principal Padé family introduced in [17] and discussed in [8, 10, 11, 4]. We prove that the family can be obtained by the König root-finding method applied to the polynomial x^2-1, which goes back to a work of Schröder in 1870 [20]. Second, using the characterization given above, we extend to the König family for the polynomial x^p-1 a result of Higham et al. [10] concerning the property of a part of the principal Padé family of preserving the structure of group of automorphisms associated with a bilinear or a sesquilinear form. Third, we show that the Halley method, which belongs to the König family, for the computation of the principal pth root of a matrix preserves the structure described above, and we prove a result on the convergence of that method. Finally, we show that the idea of the Schur–Newton method proposed by Guo and Higham in [6] for the inverse Newton iteration for the computation of the principal pth root of a matrix can be applied to the direct Newton iteration and to the Halley method, providing new algorithms with good numerical properties.

We recall that the principal pth root of a matrix A having no nonpositive real eigenvalues is the unique solution X of the matrix equation $X^p - A = 0$, such that the eigenvalues of X have argument less in modulus than π/p.

The paper is organized as follows: in section 2 we define the class of pure rational matrix iterations and we discuss their convergence; in section 3 we show the equivalence between the principal Padé iterations and the König iterations for x^2-1; in section 4 we generate a König family of matrix iterations preserving the structure of a group of automorphisms; in sections 5 and 6 we prove convergence results for the Newton and Halley method, and we extend the idea of the Schur–Newton method of Guo and Higham to them.

2. Pure rational matrix iterations. Given a rational function φ, the iteration

$$\begin{cases} z_0 \in \mathbb{C}, \\ z_{k+1} = \varphi(z_k), \quad k = 0, 1, 2, \dots, \end{cases} \tag{2.1}$$

is called a rational iteration. The function φ can have poles, so that the sequence is not necessarily well defined for each z_0. We use the notation $\varphi^{\circ k}$ to denote the kth iterate of the function φ, i.e., $\varphi^{\circ 1} = \varphi$ and $\varphi^{\circ k+1} = \varphi \circ \varphi^{\circ k}$. A fixed point z_* of (2.1) is such that $\varphi(z_*) = z_*$ and is said to be *attractive* if $|\varphi'(z_*)| < 1$. For an attractive fixed point z_*, the *basin of attraction* is the set $\mathcal{B} = \{z_0 \in \mathbb{C} : z_k \to z_*\}$; the *immediate basin* is the connected component of $\mathcal{B}$ which contains z_*.

We state a useful lemma on the basin of attraction which is a special case of Theorem 6.3.1 of [1].

LEMMA 2.1. *Let z_* be an attractive fixed point of iteration* (2.1). *The sequence $\varphi^{\circ k}(z)$ converges locally uniformly to z_* for each z_0 belonging to the basin of z_*. In other words, z_0 has a neighborhood in which $\varphi^{\circ k}$ converges uniformly to z_*.*

Proof. Since $|\varphi'(z_*)| < 1$, there exists a closed disk D centered at z_* and such that $|\varphi(z) - z_*| \leqslant M|z - z_*|$ for a positive constant $M < 1$ and for each $z \in D$, and thus $\varphi^{\circ k}$ converges uniformly on the compact sets of D.

Let z_0 belong to the basin of attraction of z_*. There exists m such that $\varphi^{\circ m}(z_0)$ belongs to the interior of D. Since $\varphi^{\circ m}$ is continuous, there exists a compact neighborhood K of z_0 such that $\varphi^{\circ m}(z)$ is a compact set fully contained in the interior of D, and thus $\varphi^{\circ k}(z)$ converges uniformly to z_* for each $z \in K$. ∎

In the matrix case, a formula like (2.1) would give an iteration of the form

$$\begin{cases} Z_0 \in \mathbb{C}^{n\times n}, \\ Z_{k+1} = \varphi(Z_k), \quad k = 0, 1, 2, \dots, \end{cases} \tag{2.2}$$

where $\varphi(z)$ is a rational function and $\varphi(Z)$, where Z is a square matrix, is defined by substituting Z for z and replacing scalar numbers by multiples of the identity and arithmetic operations by matrix operations. That procedure leads to the usual definition of function of a matrix [12, 5, 9]. We call an iteration defined by a function, as in (2.2), a *pure rational matrix iteration*.

The class of pure rational matrix iterations is not suitable to approximate generic matrix functions, since, as we will explain in Remark 2.5, there hold strong conditions on the limits of such sequences.

A larger class of iterations than the pure rational matrix iterations can be studied with similar techniques. An iteration in this larger class can be written in the form

$$\left\{ \begin{array}{l} Z_0 = p(A), \\ Z_{k+1} = \psi(Z_k, A), \quad k = 0, 1, 2, \ldots, \end{array} \right. \tag{2.3}$$

where A is a square matrix, $\psi = \psi(t, z)$ is a two-variable rational function, and p is a polynomial. In that case, for each A, the sequence Z_k defines the same sequence of rational functions $s_k(z)$ such that $s_k(A) = \psi(s_{k-1}(A), A) = Z_k$ and $s_0(A) = p(A)$. That class contains the pure rational matrix iterations as a special case if p is the identity function and the formula for ψ does not contain A.

Consider an iteration of the class (2.3) described above. Let Z_k be the kth iterate, so that $Z_k = s_k(A)$, with $s_k(z)$ being a rational function. Using the Jordan canonical form of A, say, $M^{-1}AM = J_1 \oplus \cdots \oplus J_r$, one has $M^{-1}Z_kM = s_k(J_1) \oplus \cdots \oplus s_k(J_r)$ for each k. Therefore, by means of similarity M, the iteration can be uncoupled into r iterations involving only functions of the Jordan blocks. The study of the convergence is thus restricted to the case in which A is a Jordan block of arbitrary size for the eigenvalue λ, which will be denoted by J.

Moreover, in view of the formula for a function of a Jordan block [5, Thm. 11.1.1],

$$f(J) = \begin{bmatrix} f(\lambda) & f'(\lambda) & \ldots & \frac{f^{(k-1)}(\lambda)}{(k-1)!} \\ & f(\lambda) & \ddots & \vdots \\ & & \ddots & f'(\lambda) \\ \bigcirc & & & f(\lambda) \end{bmatrix}, \tag{2.4}$$

each of the iterates is upper triangular.

A question arises naturally: if the sequence $s_k(\lambda)$, with $s_0(\lambda) = p(\lambda)$, converges for each eigenvalue of A, what can be said about the convergence of $s_k(A)$? The following easy example shows that, in general, scalar convergence does not imply matrix convergence.

Example 2.2. Consider the rational iteration $z_{k+1} = \varphi(z_k)$, where $\varphi(z) = z^2$. The sequence $\varphi^{\circ k}(1)$ converges to 1, but it fails to converge uniformly on any neighborhood of the point 1. Consider the matrix iteration $Z_{k+1} = Z_k^2$ and the starting point $Z_0 = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$; the iterates are $Z_k = \left[\begin{smallmatrix} 1 & 2^k \\ 0 & 1 \end{smallmatrix}\right]$, and the sequence fails to converge. For this iteration and Z_0 being a Jordan block of size n for the eigenvalue 1, there is matrix convergence only for $n = 1$, that is, in the scalar case.

A sufficient condition for the convergence of the matrix sequence, given the scalar convergence, is stated in Lemma 2.3, in which the notation $\|f(z)\|_K = \sup_{x \in K} |f(x)|$ is used for a function f defined on a compact set K. This approach generalizes a proof of matrix convergence in [15]. A different approach for the matrix convergence has been used in [17] and generalized in [9, Thm. 4.15], where it is proved that the matrix

convergence follows from the scalar convergence of the eigenvalues to attracting fixed points.

LEMMA 2.3. *If $s_k(z)$ is a sequence of rational functions that converges uniformly in a compact neighborhood K of λ to the function $f(z)$, then $s_k(J)$ converges to $f(J)$, where J is a Jordan block of arbitrary size n relative to the eigenvalue λ. Moreover, there exists a function $c = c(n)$, independent of k, such that*

$$\|s_k(J) - f(J)\|_\infty \leqslant c\|s_k(z) - f(z)\|_K. \tag{2.5}$$

Proof. The function f is holomorphic on K since it is the uniform limit on a compact set of holomorphic functions.

From formula (2.4), the matrix sequence converges if the sequence $s_k(z)$ and its derivatives up to the order $n-1$ converge. Consider a small circle γ of radius R, centered at λ and fully contained in K. Using the Cauchy formula, for $p = 0, 1, \ldots, n-1$, it holds that

$$\left|\frac{s_k^{(p)}(\lambda)}{p!} - \frac{f^{(p)}(\lambda)}{p!}\right| = \left|\frac{1}{2\pi i}\oint_\gamma \frac{s_k(z) - f(z)}{(z-\lambda)^{p+1}}dz\right| \leqslant \frac{1}{R^p}\|s_k(z) - f(z)\|_K.$$

The previous relation provides the convergence of the sequence $s_k(J)$ to $f(J)$, since the latter term tends to zero as k tends to ∞ by the uniform convergence assumption. It also provides the proof of (2.5), since from formula (2.4) it follows that

$$\|s_k(J) - f(J)\|_\infty = \sum_{p=0}^{n-1}\left|\frac{s_k^{(p)}(\lambda)}{p!} - \frac{f^{(p)}(\lambda)}{p!}\right| \leqslant \|s_k(z) - f(z)\|_K \sum_{p=0}^{n-1}\frac{1}{R^p}. \tag{2.6}$$ ∎

In summary, if the sequence $s_k(z)$ converges uniformly on a compact neighborhood of the spectrum of A, then the sequence $s_k(A)$ converges, and formula (2.6) can be used to provide an upper bound for the convergence of the matrix sequence. If the scalar convergence is not uniform, then the matrix iteration may fail to converge, as Example 2.2 shows.

We have turned the problem from matrix convergence to uniform scalar convergence on a compact neighborhood of the spectrum. This does not seem at first sight an advantage, but its benefit is clear in the case of pure rational iterations; in fact, Lemma 2.1 shows that if the sequence $s_k(\lambda)$ converges, for each eigenvalue λ of A, to an attractive fixed point λ_*, then the sequence $s_k(z)$ converges uniformly to λ_* on a neighborhood of λ. We have the following result.

THEOREM 2.4. *Let $Z_{k+1} = \varphi(Z_k)$ be a pure rational matrix iteration. If for each eigenvalue λ of Z_0 the scalar sequence $z_{k+1} = \varphi(z_k)$, $z_0 = \lambda$, converges to an attractive fixed point λ_*, then there exists a locally constant function $f(z)$ such that for each initial value Z in a neighborhood of Z_0 the matrix iteration converges to $f(Z)$. Moreover, $f(Z)$ is diagonalizable.*

Proof. Lemma 2.1 guarantees that the scalar iteration converges uniformly in a compact neighborhood K of spectrum of Z_0 to a locally constant function $f(z)$. Lemma 2.3 provides the matrix convergence for the Jordan blocks relative to eigenvalues belonging to the interior of K. Since the eigenvalues of a matrix are continuous functions of the entries, there exists a neighborhood V of Z_0 in the space of square matrices such that for each matrix Z of V, the eigenvalues of Z belong to the interior of K, so the matrix iteration converges to $f(Z)$.

The diagonalizability follows from the fact that f is locally constant and thus its derivatives are 0: By formula (2.4), $f(J)$ is a diagonal matrix for each Jordan block J. ∎

Remark 2.5. Theorem 2.4 states that the limit of a pure rational matrix iteration is a (scalar) locally constant function of the initial value, and it is diagonalizable provided that the convergence of the scalar sequence on the eigenvalues of Z_0 is to attractive fixed points (a scalar locally constant function need not be locally constant if applied to matrices; consider, for instance, the matrix sign function). A consequence is that only (scalar) locally constant functions can be the limit of a pure rational matrix iteration, which explains why in the literature the sole matrix functions computed using pure rational matrix iterations are the matrix sign function and the matrix sector function, which are (scalar) locally constant.

On the other hand, Theorem 2.4 shows that a function which is not (scalar) locally constant cannot be the limit of a pure rational matrix iteration; thus there is no hope to find, for instance, a pure rational matrix iteration converging to the matrix pth root, logarithm or exponential.

We note that Theorem 2.4 implies that a matrix function defined as the limit of a pure rational iteration is diagonalizable, which in particular gives another proof of the diagonalizability of the matrix sign function.

Remark 2.6. A convergence result for iterations of the type (2.3) is given by Higham [9, Thm. 4.15], generalizing a result for the matrix sign function in [17, Lem. 5.1]. His result guarantees matrix convergence if the scalar eigenvalue sequences converge to attractive fixed points. When [9, Thm. 4.15] is specialized to the case of pure rational matrix iterations, it gives a result similar to, but weaker than, Theorem 2.4. Theorem 2.4, together with Lemma 2.3, has the advantage of specifying the limit to which the matrix sequence converges as a matrix function, provides a bound for the matrix convergence, and can be further extended to the case $|\varphi'(\lambda^*)| = 1$ using, for instance, the Leau–Fatou theorem [1].

3. Equivalence between the König family and the principal Padé iterations family. In the paper [17] Kenney and Laub derive a family of rational iterations for the computation of the matrix sign function. The derivation is based on the theory of Padé approximations and exploits the relation

$$\operatorname{sign}(z) = \frac{z}{\sqrt{z^2}} = \frac{z}{\sqrt{1-(1-z^2)}} = \frac{z}{\sqrt{1-\xi}},$$

where $\xi = 1 - x^2$. They consider the approximants of the function

$$h(\xi) = (1-\xi)^{-1/2},$$

which are well known.

Given $p_{mn}(\xi)/q_{mn}(\xi)$, the (m,n) Padé approximant to h, the recurrence

$$x_{k+1} = f_{mn}(x_k) = x_k \frac{p_{mn}(1-x_k^2)}{q_{mn}(1-x_k^2)},$$

defines a family of iterations for the matrix sign function.

The iterations with $m = n-1$ and $m = n$ are globally convergent and have been called *principal Padé iterations* [9]. For these values of m and n one can define

$$g_r(x) = f_{mn}(x) \text{ for } r = m+n+1, \tag{3.1}$$

for which we have the following result [17].

THEOREM 3.1. *For the function* (3.1) *it holds that*

1. *for each nonimaginary* x_0*, the iteration* $x_{k+1} = g_r(x_k)$ *is convergent to* $\operatorname{sign}(x_0)$*, with order of convergence* r*; and*
2. $g_r(x) = \frac{(1+x)^r-(1-x)^r}{(1+x)^r+(1-x)^r}$.

Higham [9] noticed that these families were essentially derived by Howland [14]; though for even r the iteration functions of Howland are the reciprocal of those of Kenney and Laub.

In fact, the family of principal Padé iterations is a particular case of iterations going back to Schröder in his monumental paper of 1870 [20] (an English translation is available in [21]). This family was studied by Householder [13] and many other authors, who called it the König family [3] or the basic family [16].

The König method of order σ, applied to the function f, is defined by the formula [3]

$$K_{f,\sigma}(z) = z + (\sigma-1)\frac{(1/f(z))^{(\sigma-2)}}{(1/f(z))^{(\sigma-1)}}, \tag{3.2}$$

where $(1/f)^{(k)}$ is the kth derivative of $1/f$. It can be proved that the method converges to simple roots of f with order at least σ. For $\sigma = 2$ the König method is the Newton method, while for $\sigma = 3$ it is the so-called Halley method.

If f is a polynomial, then $K_{f,\sigma}$ is a rational function. Let us define $K_{p,\sigma}$ as the König family applied to the polynomial $f = x^p - 1$.

THEOREM 3.2. *For the König rational functions relative to the polynomial* x^2-1 *it holds that* $K_{2,r}(x) = \frac{(x+1)^r+(x-1)^r}{(x+1)^r-(x-1)^r}$. *Thus,* $K_{2,r}$ *coincides with* g_r *of* (3.1) *for odd* r *and with the reciprocal of* g_r *for even* r.

Proof. From

$$\frac{d^n}{dx^n}\left(\frac{1}{x^2-1}\right) = \frac{1}{2}\frac{d^n}{dx^n}\left(\frac{1}{x-1}-\frac{1}{x+1}\right)$$
$$= \frac{(-1)^n n!}{2}\left(\frac{1}{(x-1)^{n+1}}-\frac{1}{(x+1)^{n+1}}\right) = \frac{(-1)^n n!}{2}\left(\frac{(x+1)^{n+1}-(x-1)^{n+1}}{(x^2-1)^{n+1}}\right),$$

it follows that

$$K_{2,r}(x) = x - (x^2-1)\frac{(x+1)^{r-1}-(x-1)^{r-1}}{(x+1)^r-(x-1)^r} = \frac{(x+1)^r+(x-1)^r}{(x+1)^r-(x-1)^r}. \qquad \square$$

4. Structure-preserving algorithms in the König family. It has been proved in [11, Thm. 3.13] that an iteration of the form

$$z\frac{q(z)}{\operatorname{rev} q(z)}, \tag{4.1}$$

where $\operatorname{rev} q(z) = z^d q(1/z)$ for a real polynomial $q(z)$ of degree d, preserves the structure of group of automorphisms associated with

- a bilinear form on $\mathbb{R}^n$ or $\mathbb{C}^n$ and
- a sesquilinear form on $\mathbb{C}^n$.

To ease the notation we call *structure-preserving* an iteration with the form (4.1), recalling that the rational functions preserving bilinear or sesquilinear forms (fully characterized in [11, Thm. 3.13]) are more general.

The principal Padé iterations and, in view of Theorem 3.2, the $K_{2,\sigma}$ iterations for odd σ are iterations for the matrix sign function which are structure-preserving [11]; this is a case of a more general theorem.

THEOREM 4.1. *If $n \equiv 3 \pmod p$, then the function $K_{p,n}$, namely, the König method for the polynomial $x^p - 1$, has the form*

$$z\frac{q(z^p)}{\operatorname{rev} q(z^p)}, \tag{4.2}$$

where q is a real polynomial, and so in particular it is structure-preserving.

Proof. The proof is obtained by deriving a formula for the derivatives of $1/(x^p-1)$ and, from it, an explicit elementary formula for the König function from which we deduce the theorem. Let $p \geqslant 3$; the case $p = 2$ follows easily from Theorem 3.2.

Let $\omega = \cos(2\pi/p) + \mathbf{i}\sin(2\pi/p)$ and $\varphi(x) = (x^p-1)/(x-1) = \sum_{k=0}^{p-1} x^k$. Observe that $\varphi(\omega^k) = 0$ for $k \not\equiv 0 \pmod p$.

It holds that

$$\frac{1}{x^p-1} = \frac{1}{p}\sum_{k=0}^{p-1}\frac{\omega^k}{x-\omega^k};$$

in fact,

$$\begin{aligned}\sum_{k=0}^{p-1}\frac{\omega^k}{x-\omega^k} &= \frac{1}{x^p-1}\sum_{k=0}^{p-1}\frac{1}{\bar\omega^k}\frac{x^p-1}{x-\omega^k} = \frac{1}{x^p-1}\sum_{k=0}^{p-1}\frac{(\bar\omega^k x)^p-1}{\bar\omega^k x-1}\\ &= \frac{1}{x^p-1}\sum_{k=0}^{p-1}\varphi(\bar\omega^k x) = \frac{1}{x^p-1}\sum_{k=0}^{p-1}\sum_{r=0}^{p-1}(\bar\omega^k x)^r = \frac{1}{x^p-1}\sum_{r=0}^{p-1}x^r\sum_{k=0}^{p-1}\bar\omega^{kr} = \frac{p}{x^p-1}.\end{aligned}$$

Now,

$$\begin{aligned}\frac{d^n}{dx^n}\left(\frac{1}{x^p-1}\right) &= \frac{1}{p}\sum_{k=0}^{p-1}\frac{d^n}{dx^n}\frac{\omega^k}{x-\omega^k} = \frac{(-1)^n n!}{p}\sum_{k=0}^{p-1}\frac{\omega^k}{(x-\omega^k)^{n+1}}\\ &= \frac{(-1)^n n!}{p(x^p-1)^{n+1}}\sum_{k=0}^{p-1}\bar\omega^{kn}\varphi^{n+1}(\bar\omega^k x) = \frac{(-1)^n n!}{p(x^p-1)^{n+1}}\sum_{k=0}^{p-1}\omega^{kn}\varphi^{n+1}(\omega^k x),\end{aligned}$$

and defining $\psi_n(x) = \frac{1}{p}\sum_{k=0}^{p-1}\omega^{k(n-1)}\varphi^n(\omega^k x)$ yields the explicit formula

$$K_{p,n} = x - (x^p-1)\frac{\psi_{n-1}}{\psi_n} = \frac{x\psi_n - (x^p-1)\psi_{n-1}}{\psi_n}.$$

The denominator of $K_{p,n}$, namely, $\psi_n(x)$, is formed by the terms of $\varphi^n(x)$ in which the exponent of x is congruent to $(1-n)$ modulo p; in fact, if $\varphi^n(x) = \sum a_r x^r$, then

$$\psi_n(x) = \frac{1}{p}\sum_{k=0}^{p-1}\omega^{k(n-1)}\sum_r a_r\omega^{kr}x^r = \frac{1}{p}\sum_r\left(a_r x^r\sum_{k=0}^{p-1}\omega^{k(n+r-1)}\right) = \sum_{r\equiv 1-n} a_r x^r.$$

The numerator of $K_{p,n}$, namely, $x\psi_n(x)-(x^p-1)\psi_{n-1}(x)$, is formed by the terms of $\varphi^n(x)$ in which the exponent of x is congruent to $(2-n)$ modulo p; in fact,

$$x\psi_n(x)-(x^p-1)\psi_{n-1}(x) = \frac{1}{p}\sum_{k=0}^{p-1}\left(\omega^{k(n-1)}x\varphi^n(\omega^k x)-\omega^{k(n-2)}(x^p-1)\varphi^{n-1}(\omega^k x)\right)$$

$$= \frac{1}{p}\sum_{k=0}^{p-1}\left(\omega^{k(n-1)}x\varphi^n(\omega^k x)-\omega^{k(n-2)}(\omega^k x-1)\varphi^n(\omega^k x)\right) = \frac{1}{p}\sum_{k=0}^{p-1}\omega^{k(n-2)}\varphi^n(\omega^k x)$$

$$= \frac{1}{p}\sum_{k=0}^{p-1}\omega^{k(n-2)}\sum_r a_r\omega^{kr}x^r = \frac{1}{p}\sum_r\left(a_r x^r\sum_{k=0}^{p-1}\omega^{k(n+r-2)}\right) = \sum_{r\equiv 2-n} a_r x^r,$$

where we have used the identity $x^p-1=(\omega^k x-1)\varphi(\omega^k x)$ for any k.

Let $a_{\alpha_1},\dots,a_{\alpha_\nu}$ be the coefficients of $\varphi^n(x)$ relative to exponents congruent to $1-n$ modulo p, and let $a_{\beta_1},\dots,a_{\beta_\mu}$ be the coefficients of $\varphi^n(x)$ relative to exponents congruent to $2-n$ modulo p, so that

$$K_{p,n} = \frac{a_{\beta_1}x^{\beta_1}+\cdots+a_{\beta_\mu}x^{\beta_\mu}}{a_{\alpha_1}x^{\alpha_1}+\cdots+a_{\alpha_\nu}x^{\alpha_\nu}}.$$

To conclude the proof, it is enough to prove that, for $n\equiv 3 \pmod p$, it holds that $\mu=\nu$ and $a_{\alpha_1}=a_{\beta_\mu}, a_{\alpha_2}=a_{\beta_{\mu-1}},\dots,a_{\alpha_\nu}=a_{\beta_1}$.

Let $N=\deg\varphi^n(x)=np-n$. To prove the equality $\mu=\nu$ observe that μ and ν are the numbers of solutions of the congruences $r\equiv 1-n \pmod p$ and $r\equiv 2-n \pmod p$, respectively, such that $0\leqslant r\leqslant N$. For $n\equiv 3 \pmod p$ there exists an integer γ such that $N=\gamma p-3$; thus the number of solutions of the two congruences $r\equiv 1-n\equiv -2 \pmod p$ and $r\equiv 2-n\equiv -1 \pmod p$ such that $0\leqslant r\leqslant N$ is the same.

Observe that since $N=np-n$, then $\beta_\mu=N+2-p$, and observe that $\varphi^n(x)=\operatorname{rev}\varphi^n(x)$, namely, $a_r=a_{N-r}$ for each $r=0,1,\dots$. For $n\equiv 3 \pmod p$, it holds that $\alpha_1=p-2$, and thus $a_{\alpha_1}=a_{p-2}=a_{N+2-p}=a_{\beta_\mu}$.

The equalities $a_{\alpha_{i+1}}=a_{\beta_{\mu-i}}$ for $i=1,2,\dots$ follow from the fact that if $n\equiv 3 \pmod p$, then $\alpha_{i+1}=(i+1)p-2=N-(N+2-p-ip)=N-\beta_{\mu-i}$.

Simplifying the common factors gives the required form for $K_{p,n}$. □

By the properties of the König method [3], the iteration $z_{k+1}=K_{p,n}(z_k)$ converges locally, with order of convergence at least n, to the roots of the polynomial x^p-1. It is easy to see, by an induction argument, that the iteration

$$x_{k+1}=\zeta K_{p,n}(\zeta^{-1}x_k), \tag{4.3}$$

where ζ is any pth root of the nonzero scalar a, for $x_0=\zeta z_0$, is such that $x_k=\zeta z_k$ and thus converges locally to the roots of $x^p-a=0$.

Iteration (4.3) does not seem effective for computing the pth roots of a, since it uses ζ, but for $n\equiv 3 \pmod p$; in view of Theorem 4.1, the iteration for x_k has the form

$$x_{k+1}=x_k\frac{q(a^{-1}x_k^p)}{\operatorname{rev}q(a^{-1}x_k^p)}=x_k\frac{\widehat{q}(x_k^p)}{\operatorname{rev}\widehat{q}(x_k^p)}, \tag{4.4}$$

where $\widehat{q}$ is obtained by multiplying q by a suitable power of a. In this way, an effective iteration is obtained to approximate with high precision the pth roots of a given complex number.

A difficulty in the use of iteration (4.4) is the global convergence. We will not investigate further the global convergence of (4.4), but in section 5 we will give a convergence proof for the case $n = 3$, which is a structure-preserving iteration for each p, in view of Theorem 4.1.

Remark 4.2. Theorem 4.1 has a perhaps surprising application to the theory of root-finding algorithms. Following McMullen [19], a rational iterative root-finding algorithm is said to be *generally convergent* if it converges to a root for almost every initial guess and for almost every polynomial (where the Lebesgue measure on the complex plane and on the space of coefficients is considered).

It is known that the Newton method is generally convergent for quadratic polynomials, but not for cubics. In fact, the Newton iteration for the polynomial $p(z) = z^3 - 2z + 2$ does not converge to any root for initial values in a suitable set of measure greater than zero.

McMullen constructed in [19] a generally convergent algorithm for cubic polynomials and proved that there does not exist a generally convergent algorithm for polynomials of degree greater than 3.

Using the results of McMullen, Hawkins proved that any generally convergent root-finding algorithm is generated by a root-finding algorithm for the polynomial $x^3 - 1$ of the form (4.2) [7]. Thus, Theorem 4.1 could be used to construct generally convergent algorithms for a cubic polynomial of an arbitrarily high order of convergence.

5. Nice properties of the Halley method. The König method of order 3 is the so-called Halley method, which, for the equation $x^p - 1 = 0$, is

$$x_{k+1} = x_k \frac{(p-1)x_k^p + (p+1)}{(p+1)x_k^p + (p-1)}, \quad x_0 \in \mathbb{C}. \tag{5.1}$$

Here we considered a matrix generalization of the Halley method for computing the principal pth root of a matrix A.

A very nice feature of the Halley method for the equation $x^p - 1 = 0$ is that the basin of attraction for the fixed point 1 is somewhat *nicer* than that of the Newton method (see Figure 5.1 for a comparison in the case $p = 4$). It was proved [15] that for the Newton method applied to the equation $x^p - 1 = 0$ the basin of attraction for the fixed point 1 contains the set

$$\mathcal{T}_{2p} = \{z \in \mathbb{C} \setminus \{0\} \,:\, -\pi/(2p) < \arg(z) < \pi/(2p), |z| \geqslant 1\}, \tag{5.2}$$

while for the Halley method there holds the following result.

THEOREM 5.1. *The immediate basin of attraction for the fixed point* 1 *of the rational iteration* (5.1) *contains the sector*

$$\mathcal{S}_{2p} = \{z \in \mathbb{C} \setminus \{0\} \,:\, -\pi/(2p) < \arg(z) < \pi/(2p)\}. \tag{5.3}$$

Proof. Let us define

$$\varphi(z) = \frac{(p-1)z^p + (p+1)}{(p+1)z^p + (p-1)};$$

iteration (5.1) can be written as $z_{k+1} = z_k\varphi(z_k)$. The sector $\mathcal{S}_{2p}$ contains the fixed point $z = 1$ and is an open connected set, and, by Lemma 5.2, if $z \in \mathcal{S}_{2p}$, then $z\varphi(z) \in \mathcal{S}_{2p}$. Thus, the set $\mathcal{S}_{2p}$ belongs to the immediate basin of the fixed point $z = 1$. In fact, given a rational iteration $x_{k+1} = \psi(x_k)$ of degree greater than 1,

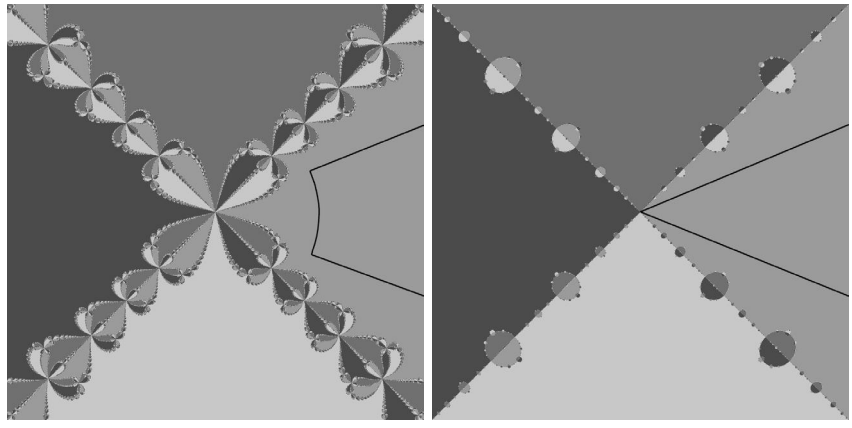


FIG. 5.1. *Comparison of the basins of attraction for the Newton method (left) and the Halley method (right) for the equation* $x^4 - 1$ *in the set* $[-2,2] \times [-2,2] \subset \mathbb{C}$. *The sets* $\mathcal{T}_{2p}$ *of* (5.2) *and* $\mathcal{S}_{2p}$ *of* (5.3) *are highlighted.*

any connected open set $\mathcal{U}$ such that $\psi(\mathcal{U}) \subset \mathcal{U}$ and containing only a fixed point z_* belongs to the immediate basin of z_* (compare with [1, Thm. 4.2.5]). □

LEMMA 5.2. *For each* $z \in \mathcal{S}_{2p}$, *it holds that* $|\arg(z\varphi(z))| \leqslant |\arg(z)|$ *and the equality holds if and only if* z *is real.*

Proof. If z is real, then $\varphi(z)$ is real. Let us consider the case $\arg(z) > 0$; since $\arg(z\varphi(z)) = \arg(z) + \arg(\varphi(z))$, it is enough to prove that

$$-2\arg(z) < \arg(\varphi(z)) < 0. \tag{5.4}$$

Removing real positive constants, it holds that

$$\arg(\varphi(z)) = \arg\big(((p-1)z^p + (p+1)) \cdot ((p+1)\bar{z}^p + (p-1))\big).$$

Using the decomposition $z = r(\cos\vartheta + \mathbf{i}\sin\vartheta)$, one has

$$\arg(\varphi(z)) = \arg\big((p^2-1)(|r|^{2p}+1) + 2(p^2+1)r^p\cos(p\vartheta) - 4\mathbf{i}pr^p\sin(p\vartheta)\big).$$

Applying the tangent trigonometric function to the inequalities (5.4), we obtain the equivalent

$$-\frac{\sin(2\vartheta)}{\cos(2\vartheta)} < \frac{-4pr^p\sin(p\vartheta)}{(p^2-1)(r^{2p}+1) + 2(p^2+1)r^p\cos(p\vartheta)} < 0. \tag{5.5}$$

The latter inequality is evident from $0 < \vartheta < \pi/(2p)$. The former needs a bit more work; it can be rewritten as

$$\begin{aligned}(p^2-1)\sin(2\vartheta)r^{2p} + 2\big((p^2+1)\cos(p\vartheta)\sin(2\vartheta) - 2p\sin(p\vartheta)\cos(2\vartheta)\big)r^p \\ + (p^2-1)\sin(2\vartheta) > 0\end{aligned} \tag{5.6}$$

and can be seen as a quadratic inequality on the variable $x = r^p$. The quadratic has the form $\gamma(x) = ax^2 + 2bx + a$, where $a = (p^2-1)\sin(2\vartheta)$ and $b = (p^2+1)\cos(p\vartheta)\sin(2\vartheta) - 2p\sin(p\vartheta)\cos(2\vartheta)$. Since $a > 0$, the inequality $\gamma(x) > 0$ is true if the equation $\gamma(x) = 0$ has no solution. Observe that if $\gamma(x) = 0$, then $\gamma(1/x) = 0$, and if $\gamma(1) > 0$, there exists no positive solution.

Using the inequalities $(\vartheta - \vartheta^3/6) \leqslant \sin\vartheta \leqslant \vartheta$ for $0 < \vartheta < \pi/p$ and $\sin((p-2)\vartheta) = \sin(p\vartheta)\cos(2\vartheta) - \sin(2\vartheta)\cos(p\vartheta)$, one can see that

$$\frac{1}{2}\gamma(1) \geqslant (p^2-1)\sin(2\vartheta) - 2p\sin((p-2)\vartheta) > (p^2-1)\left(2\vartheta - \frac{4}{3}\vartheta^3\right) - 2p(p-2)\vartheta$$
$$= \frac{2}{3}\vartheta\left(6p - 3 - 2(p^2-1)\vartheta^2\right);$$

the last expression is positive if $\vartheta^2 \leqslant \frac{6p-3}{2p^2-2}$, and this is true since $\vartheta \leqslant \pi/(2p)$. ∎

It is worth giving a corollary of Theorem 5.1 which could be used for the computation of the scalar pth root.

COROLLARY 5.3. *Consider the Halley method for the equation* $x^p - a = 0$,

$$x_{k+1} = x_k \frac{(p-1)x_k^p + (p+1)a}{(p+1)x_k^p + (p-1)a}, \quad x_0 \in \mathbb{C}. \tag{5.7}$$

The principal basin for the initial value $x_0 = 1$ *contains the set* $\mathbb{C}_> = \{z \in \mathbb{C} : \operatorname{Re} z > 0\}$.

Theorem 2.4 guarantees the convergence of the pure matrix iteration

$$Y_{k+1} = Y_k((p-1)Y_k^p + (p+1)I)((p+1)Y_k^p + (p-1)I)^{-1} \tag{5.8}$$

to the identity matrix I for each Y_0 having eigenvalues in $\mathcal{S}_{2p}$, particularly for $Y_0 = A^{-1/p}$, where A has eigenvalues in the open right half complex plane, which will be denoted by $\mathbb{C}_>$. Iteration (5.8) is strictly related to

$$X_{k+1} = X_k((p-1)X_k^p + (p+1)A)((p+1)X_k^p + (p-1)A)^{-1}; \tag{5.9}$$

in fact, if A has eigenvalues in $\mathbb{C}_>$, $Y_0 = A^{-1/p}$, and $X_0 = I$, then it can be shown that $X_k = Y_k A^{1/p}$ for each k (the proof follows by an induction argument and using the fact that X_k and Y_k are functions of A and so commute with A and $A^{1/p}$).

COROLLARY 5.4. *The sequence* X_k *obtained by iteration* (5.9) *with* $X_0 = I$ *converges to* $A^{1/p}$ *for each* A *having eigenvalues in* $\mathbb{C}_>$.

Moreover, for what we have proved in section 4, iteration (5.8) is structure-preserving. If A belongs to a group of automorphisms as in section 4, so does $A^{1/p}$; thus, each of the iterates obtained by (5.9) belongs to that group.

Iteration (5.9) cannot be used directly to approximate the principal pth root. In fact, using the same idea as in [15], one can prove that iteration (5.9) is *not stable in a neighborhood of* $A^{1/p}$; i.e., a perturbation on the value of X_k is amplified in the following steps preventing the convergence in a finite arithmetic computation.

This problem can be overridden using another algorithm which provides the same sequence but which is stable in a neighborhood of $A^{1/p}$—for instance,

$$\begin{cases} X_0 = I, \quad N_0 = A, \\ X_{k+1} = X_k((p+1)I + (p-1)N_k)^{-1}((p-1)I + (p+1)N_k), \\ N_{k+1} = N_k\left(((p+1)I + (p-1)N_k)^{-1}((p-1)I + (p+1)N_k)\right)^{-p}, \end{cases} \tag{5.10}$$

where $N_k \to I$ and $X_k \to A^{1/p}$. If the pth power is computed using the binary powering technique [5, Alg. 11.2.2], the computational cost of iteration (5.10) is $2(5 + \vartheta\log_2 p)n^3$ arithmetic operations (ops) per step, where $1 \leqslant \vartheta \leqslant 2$.

6. New algorithms for the matrix *p*th root. A family of iterations for computing the principal pth root of a matrix A is

$$X_{k+1} = \frac{(p-1)X_k + AX_k^{1-p}}{p}, \tag{6.1}$$

which coincides with the Newton method for the equation $X^p - A = 0$, when the latter is well defined and X_0 commutes with A [22]; this is the reason why iteration (6.1) is referred to, somehow improperly, as the Newton method.

In [22] it was proved that this iteration is not stable in a neighborhood of $A^{1/p}$. A stable variant, for $X_0 = I$,

$$\begin{cases} Y_0 = I, \quad N_0 = A, \\ Y_{k+1} = Y_k \left(\dfrac{(p-1)I + N_k}{p} \right), \\ N_{k+1} = \left(\dfrac{(p-1)I + N_k}{p} \right)^{-p} N_k, \end{cases} \tag{6.2}$$

was proposed in [15], where it was proved that (Y_k, N_k) converges quadratically to $(A^{1/p}, I)$ for each A having eigenvalues in the set

$$\mathcal{D} = \{z \in \mathbb{C} : \operatorname{Re} z > 0, |z| \leqslant 1\}. \tag{6.3}$$

This leads to an algorithm for computing the principal pth root.

Algorithm 1 (a Newton method for $A^{1/p}$ [15]). Given are $A \in \mathbb{C}^{n\times n}$ with no nonpositive real eigenvalues, an integer $p > 2$, and an algorithm for computing the square root.

1. Compute B, the principal square root of A;
2. set $C = B/\|B\|$ for a suitable norm; the eigenvalues of C belong to the set $\mathcal{D}$ of (6.3);
3. by means of iteration (6.2),
   - if p is even, compute $S = C^{2/p}$, the $(p/2)$th root of C, and set $X = S\|B\|^{2/p}$;
   - if p is odd, compute $S = C^{1/p}$, the pth root of C, and set $X = \left(S\|B\|^{1/p}\right)^2$.

Iteration (6.2) of Algorithm 1 has a computational cost of $2(3+\vartheta \log_2 p)n^3$ ops per step, where $1 \leqslant \vartheta \leqslant 2$. The initial square root can be obtained by forming the Schur decomposition of A, without affecting the complexity order with respect to p. An observation of Guo and Higham is that the Schur decomposition gives the eigenvalues of A, and that information is not exploited in Algorithm 1.

Since the number of steps to achieve the required accuracy in the numerical computation depends on the localization of the eigenvalues of the matrix whose pth root is required, a smarter preprocessing could reduce the number of steps needed for the expensive iteration (6.2) (or other similar iterations) to verify a suitable stopping criterion. In order to give a better localization of the eigenvalues, one could perform a small number of initial square roots without affecting the order of complexity of the overall algorithm. Moreover, multiplying the preprocessed matrix by a scalar parameter could further reduce the number of steps needed for convergence.

The *Schur–Newton method*, an algorithm of Guo and Higham [6], is based on these ideas. The algorithm does not use iteration (6.2) but an iteration which generalizes

the scalar Newton method for the equation $x^{-p} - a = 0$. The iteration, introduced in [2], is

$$X_{k+1} = \frac{1}{p}\left((p+1)X_k - X_k^{p+1}A\right), \quad X_0 = I, \tag{6.4}$$

which converges to $A^{-1/p}$, and for which in [6] is constructed a convergence region for the eigenvalues of A: if the spectrum of A belongs to that region, then $X_k \to A^{-1/p}$. From iteration (6.4) can be obtained a stable iteration [15, 18, 6]

$$\begin{cases} Y_0 = \dfrac{1}{c}I, \quad N_0 = \dfrac{1}{c^p}A, \\ Y_{k+1} = Y_k\left(\dfrac{(p+1)I - N_k}{p}\right), \\ N_{k+1} = \left(\dfrac{(p+1)I - N_k}{p}\right)^p N_k, \end{cases} \tag{6.5}$$

such that $Y_k \to A^{-1/p}$ and $N_k \to I$. Setting $X_k = Y_k^{-1}$ gives the iteration [6]

$$\begin{cases} X_0 = cI, \quad N_0 = \dfrac{1}{c^p}A, \\ X_{k+1} = \left(\dfrac{(p+1)I - N_k}{p}\right)^{-1} X_k, \\ N_{k+1} = \left(\dfrac{(p+1)I - N_k}{p}\right)^p N_k, \end{cases} \tag{6.6}$$

for which $X_k \to A^{1/p}$. The computational costs of iterations (6.5) and (6.6) are $2(2 + \vartheta \log_2 p)n^3$ and $2(3 + \vartheta \log_2 p)n^3$ ops per step, respectively, where $1 \leqslant \vartheta \leqslant 2$.

Algorithm 2 (Schur–Newton algorithm for $A^{1/p}$ using (6.5) and (6.6) [6]). Given are $A \in \mathbb{C}^{n\times n}$ with no nonpositive real eigenvalues and an integer $p = 2^{k_0}q$ with $k_0 \geqslant 0$ and q odd.

1. Compute the Schur decomposition of $A = QRQ^T$;
2. if $q = 1$, then $k_1 = k_0$; else choose $k_1 \geqslant k_0$ such that $\arg(\lambda_i^{1/2^{k_1}}) \in (-\pi/8, \pi/8)$ for each i and $|\lambda_1/\lambda_n|^{1/2^{k_1}} \leqslant 2$, where the eigenvalues of A are ordered $|\lambda_n| \leqslant \cdots \leqslant |\lambda_1|$;
3. compute $B = R^{1/2^{k_1}}$ by taking the square root k_1 times; if $q = 1$, then $X = QBQ^T$; else continue;
4. let $\mu_1 = |\lambda_1|^{1/2^{k_1}}$, $\mu_n = |\lambda_n|^{1/2^{k_1}}$;
   - if the λ_i are all real, if $\mu_1 \neq \mu_n$, determine $c = \left(\frac{\alpha^{1/q}\mu_1 - \mu_n}{(\alpha^{1/q}-1)(p+1)}\right)^{1/q}$ with $\alpha = \mu_1/\mu_n$; else $c = \mu_n^{1/q}$;
   - if some λ_i is complex, then $c = \left(\frac{\mu_1+\mu_n}{2}\right)^{1/q}$;
5. compute $C = B^{1/q}$ by (6.6); $X = QC^{2^{k_1-k_0}}Q^T$ (or compute $C = B^{-1/q}$ by (6.5), $X = Q(C^{2^{k_1-k_0}})^{-1}Q^T$).

The initial square roots computation, in certain cases, may dramatically reduce the number of steps needed by the iteration, but each square root in preprocessing corresponds to a squaring at the final step of the algorithm. The cost of a square root and a squaring is less than the cost of one step of the iteration, but a large number of initial square roots may result in a waste of computation if there is no saving in the number of iteration steps.

A little extension of the region of convergence D of (6.3) allows one to use the ideas of Algorithm 2 also for iteration (6.2). The proof will be given in section 6.1 and is based on the proof of Theorem 2.3 of [15].

THEOREM 6.1. *The immediate basin of attraction for the fixed point* 1 *of the iteration*

$$x_{k+1} = \frac{(p-1)x_k + x_k^{1-p}}{p} \tag{6.7}$$

contains the set

$$\mathcal{E} = \left\{ z \in \mathbb{C} : |z| \geqslant \frac{1}{2^{1/p}}, |\arg(z)| < \pi/(4p) \right\}.$$

Observe that iteration (6.1) with $X_0 = I$ converges to $A^{1/p}$ if and only if the iteration

$$X_{k+1} = \frac{(p-1)X_k + X_k^{1-p}}{p}, \quad X_0 = A^{-1/p}, \tag{6.8}$$

converges to the identity matrix. This fact and Theorem 2.4 give the following result.

COROLLARY 6.2. *Iteration* (6.1) *converges for each A having eigenvalues in*

$$D_+ = \{z \in \mathbb{C} : |z| \leqslant 2, |\arg(z)| < \pi/4\}. \tag{6.9}$$

Corollary 6.2 leads to an analogue of Algorithm 2 using iteration (6.2).

Algorithm 3 (Schur–Newton algorithm using (6.2)). Given are $A \in \mathbb{C}^{n\times n}$ with no nonpositive real eigenvalues and an integer $p = 2^{k_0}q$ with $k_0 \geqslant 0$ and q odd.

1. Compute the Schur decomposition of $A = QRQ^T$;
2. if $q = 1$, then $k_1 = k_0$; else choose $k_1 \geqslant k_0$ such that there exists a positive number s such that for each eigenvalue λ of A, $s\lambda^{1/2^{k_1}} \in \mathcal{D}$, where $\mathcal{D}$ is the disk of center 6/5 and radius 3/4;
3. compute $B = R^{1/2^{k_1}}$ by taking the square root k_1 times; if $q = 1$, then $X = QBQ^T$; else continue;
4. compute $C = (B/s)^{1/q}$ by (6.2); $X = Q(Cs^{1/q})^{2^{k_1-k_0}}Q^T$.

The convergence of Algorithm 3 is guaranteed by Corollary 6.2; in fact, iteration (6.2) is applied to a matrix having eigenvalues in the set $\mathcal{D}$, which is a subset of D_+ of (6.9). The set $\mathcal{D}$ is chosen heuristically in order to need at most 5 steps of the Newton iteration in the scalar case.

Step 2 of Algorithm 3 can be performed in an inexpensive way. For $m \geqslant k_0$ and for each eigenvalue λ of A, one looks for an interval $[t_1(\lambda), t_2(\lambda)]$ such that $t_1(\lambda) > 0$ and $t\lambda^{1/2^m}$ lies in $\mathcal{D}$ for $t \in [t_1(\lambda), t_2(\lambda)]$; if such an interval exists for each λ and the intersection is not void, then s can be any point of the intersection; else increase m.

In Figure 6.1 we have constructed experimentally the *level sets of convergence* for iterations (6.1) and (6.4) applied to scalar numbers. Given the tolerance $\varepsilon = 10^{-15}$, a point x_0 of the region $[-1, 5] \times [-3, 3]$ of the complex plane has been colored by a tonality of grey if convergence up to ε occurs in less than 10 steps. Each tonality of grey corresponds to a different number of iterations needed: the lighter one corresponds to the points for which convergence up to ε occurs in 9 steps. The black contour encloses the sets in which the eigenvalues of the matrix preprocessed by Algorithms 3 and 2 lie; observe that in the examples in Figure 6.1 the scalar iteration with an initial value

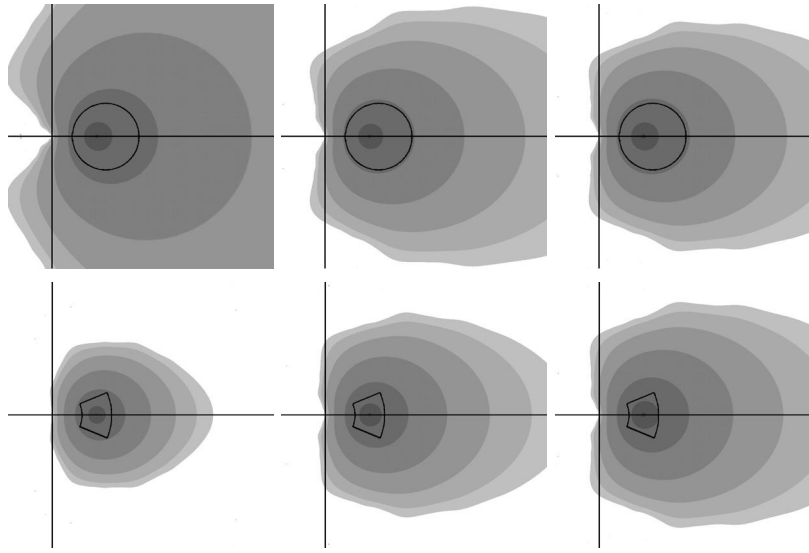


FIG. 6.1. *Comparison of the* level sets of convergence *for the Newton method (first row) and the inverse Newton method (second row) for* $p = 4$ *(first column),* $p = 20$ *(second column), and* $p = 200$ *(last column). The black contour encloses the regions in which lie the eigenvalues of the preprocessed matrices to which is applied the iterative step of Algorithms* 3 *(first row) and* 2 *(second row).*

inside the bordered regions needs at most five iterations. The expected number for the matrix iteration is the same, unless the matrix is nondiagonalizable.

In practice, due to the larger level sets of convergence (see Figure 6.1), Algorithm 3 is likely to obtain the same number of iteration steps as Algorithm 2 with a slightly milder condition on step 2, which could save a couple of square roots in preprocessing.

From the stable version of Halley's iteration (5.10) and Corollary 5.4, we obtain another algorithm.

Algorithm 4 (Schur–Halley algorithm using (5.10)). Given are $A \in \mathbb{C}^{n\times n}$ with no nonpositive real eigenvalues and an integer $p = 2^{k_0}q$ with $k_0 \geqslant 0$ and q odd.

1. Compute the Schur decomposition of $A = QRQ^T$;
2. if $q = 1$, then $k_1 = k_0$; else choose $k_1 \geqslant k_0$ such that there exists a positive number s such that for each eigenvalue λ of A, $s\lambda^{1/2^{k_1}} \in \mathcal{D}$, where $\mathcal{D}$ is the disk of center 8/5 and radius 1;
3. compute $B = R^{1/2^{k_1}}$ by taking the square root k_1 times; if $q = 1$, then $X = QBQ^T$; else continue;
4. compute $C = (B/s)^{1/q}$ by (5.10); $X = Q(Cs^{1/q})^{2^{k_1-k_0}}Q^T$.

The convergence of Algorithm 4 is guaranteed by Corollary 5.4; in fact, iteration (5.10) is applied to a matrix having eigenvalues in the set $\mathbb{C}_>$.

Once again, the choice of $\mathcal{D}$ is heuristic and is based on the observation of the experimental regions of convergence. With this preprocessing the iteration usually needs three steps to converge.

Algorithms 3 and 4 do not have the disadvantages of Algorithm 1, described in [6], i.e., a large number of steps or a possible instability in certain cases. They do have the same excellent numerical behavior of Algorithm 2; moreover, in most cases, they can save some square roots in preprocessing.

TABLE 6.1
Results for the 5th root of a random nonnormal matrix.

Algorithm 2 iteration (6.5)	Algorithm 3 iteration (6.2)	Algorithm 4 iteration (5.10)
$\rho_A(\widetilde{X}) = 3.3\text{e-}16$	$\rho_A(\widetilde{X}) = 2.7\text{e-}16$	$\rho_A(\widetilde{X}) = 2.8\text{e-}16$
$\rho_{A^{-1}}(\widetilde{X}^{-1}) = 7.4\text{e-}17$	$\rho_{A^{-1}}(\widetilde{X}^{-1}) = 4.2\text{e-}16$	$\rho_{A^{-1}}(\widetilde{X}^{-1}) = 4.7\text{e-}16$
iter=5, $k_1 = 3$	iter=5, $k_1 = 2$	iter=3, $k_1 = 2$

TABLE 6.2
Results for the 15th root of a 3-by-3 matrix A with real eigenvalues and condition number $\kappa_2(A) \approx 10^{10}$.

Algorithm 2 iteration (6.5)	Algorithm 3 iteration (6.2)	Algorithm 4 iteration (5.10)
$\text{err} = 2.7\text{e-}8$	$\text{err} = 2.7\text{e-}8$	$\text{err} = 2.7\text{e-}8$
$\rho_A(\widetilde{X}) = 5.0\text{e-}17$	$\rho_A(\widetilde{X}) = 8.1\text{e-}18$	$\rho_A(\widetilde{X}) = 1.5\text{e-}17$
iter=5, $k_1 = 5$	iter=5, $k_1 = 4$	iter=3, $k_1 = 4$

To compare the algorithms, we use the criterion used in [6], considering the *relative residual*

$$\rho_A(\widetilde{X}) \doteq \frac{\|A - \widetilde{X}^p\|}{\|\widetilde{X}\| \left\| \sum_{i=0}^{p-1} \left(\widetilde{X}^{p-1-i}\right)^T \otimes \widetilde{X}^i \right\|},$$

where $\widetilde{X}$ is the computed matrix and where the norm used is the infinity norm, and the algorithms are stopped when $\|N_k - I\| < 100nu$, where n is the size of A and u is the machine precision.

As a first test, the 5th root of a random nonnormal matrix constructed as described in [6] is computed with Algorithms 2, 3, and 4. This example was used in [6] to show the better behavior of Algorithm 2 with respect to Algorithm 1. In Table 6.1 we compare the results in terms of relative residual, number of steps (iter), and number of square roots in preprocessing (k_1).

A second test is performed considering the nonnormal matrix

$$S = \begin{bmatrix} -1 & -2 & 2 \\ -4 & -6 & 6 \\ -4 & -16 & 13 \end{bmatrix},$$

whose eigenvalues are $\{1, 2, 3\}$, and computing the 15th root of $A \doteq S^{15}$, which is formed exactly. The condition number $\kappa_2(A) = \|A\|_2\|A^{-1}\|_2$ of the matrix A is about 10^{10}. In Table 6.2 the algorithms are compared in terms of the relative residual and the relative error of the computed solution $\widetilde{X}$, namely, $\text{err} = \|\widetilde{X} - S\|/\|S\|$, where the Frobenius norm is used.

Observe that Algorithm 3 gives the same numerical results as Algorithm 2, with fewer square roots in preprocessing. Algorithm 4 requires, in general, fewer square roots in preprocessing and a minor number of steps since it has cubic convergence, though the computational cost per step is higher than in the other two. An advantage of Algorithm 4 is that it is structure-preserving.

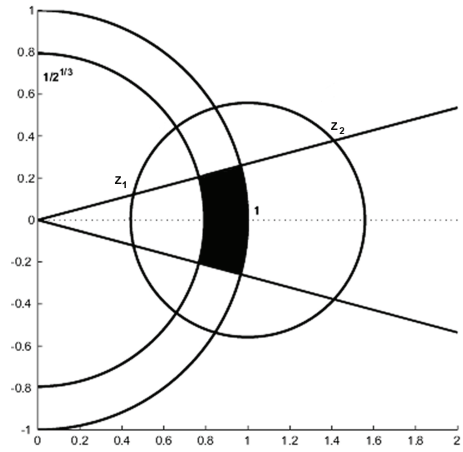


FIG. 6.2. *In black the region $\mathcal{E}$ of the proof of Theorem* 6.1 *for $p = 3$.*

6.1. Proof of Theorem 6.1. It is enough to prove that the set $\mathcal{E} \cap \{|z| < 1\}$ belongs to the immediate basin of attraction; in fact, the case $|z| \geqslant 1$ is a corollary of Theorem 2.3 of [15].

In Lemma 2.4 of [15], it is proved that a disk centered at $z = 1$ and with radius R_p is contained in the basin of 1, where $R_p = 1 - s_p$ and s_p is the unique real solution of the equation $(2p-1)s^p - 2p\,s^{p-1} + 1 = 0$ in the interval $(0, 1)$. In Lemma 2.8 of [15], it is proved that $R_p \geqslant \alpha_0/p$ for each $p > 1$, where $\alpha_0 > 1.256$.

To achieve the proof it is enough to show that the half line forming an angle of $\pi/(4p)$ with the real axis meets the circle $|z - 1| = R_p$ at two points z_1 and z_2 such that

$$r_1 < \frac{1}{\sqrt[p]{2}} < 1 < r_2,$$

where $r_1 = |z_1|$ and $r_2 = |z_2|$. That would imply that the set $\mathcal{E}$ (the black set in Figure 6.2) belongs to the disk $|z-1| \leqslant R_p$ and then to the basin of attraction of the fixed point 1.

The equation that gives the two points of intersection is $|re^{\mathbf{i}\pi/(4p)} - 1| = R_p$, which can be rewritten as

$$\gamma(r) \doteq r^2 - 2r\cos(\pi/(4p)) + 1 - R_p^2 = 0.$$

The function $\gamma(r)$ is quadratic; to prove that $r_2 > 1$, observe that

$$\gamma(1) = 2 - R_p^2 - 2\cos\left(\frac{\pi}{4p}\right) \leqslant \frac{1}{p^2}\left(\frac{\pi^2}{16} - \alpha_0^2\right) < 0.$$

The inequality $r_1 < 1/\sqrt[p]{2}$ can be written as

$$\cos(\pi/(4p)) - \sqrt{\cos^2(\pi/(4p)) - 1 + R_p^2} < \frac{1}{\sqrt[p]{2}},$$

which follows from

$$\sqrt{\cos^2(\pi/(4p)) - 1 + R_p^2} \geqslant \frac{\sqrt{\alpha_0^2 - \pi^2/16}}{p} > 0 > \frac{\log 2}{p} \geqslant \cos(\pi/(4p)) - \frac{1}{\sqrt[p]{2}},$$

where we have used the following inequalities: $\cos^2(\pi/(4p)) - 1 \geqslant -\pi^2/(16p^2)$, $R_p^2 \geqslant \alpha_0^2/p^2$, $1/\sqrt[p]{2} > 1 - \log(2)/p$, and $\cos(\pi/(4p)) < 1$. ∎

Acknowledgments. The author would like to thank Prof. Dario A. Bini, Prof. Nicholas J. Higham, and an anonymous referee whose pertinent and detailed suggestions considerably improved the presentation and the correctness of the paper.

REFERENCES

[1] A. F. Beardon, *Iteration of Rational Functions: Complex Analytic Dynamical Systems*, Grad. Texts in Math. 132, Springer-Verlag, New York, 1991.
[2] D. A. Bini, N. J. Higham, and Beatrice Meini, *Algorithms for the matrix pth root*, Numer. Algorithms, 39 (2005), pp. 349–378.
[3] X. Buff and C. Henriksen, *On König's root-finding algorithms*, Nonlinearity, 16 (2003), pp. 989–1015.
[4] A. Frommer and V. Simoncini, *Matrix functions*, in Model Order Reduction: Theory, Research Aspects and Applications, Mathematics in Industry, W. Schilders and H. A. Van der Vorst, eds., Springer-Verlag, Heidelberg, 2008, pp. 275–304.
[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[6] C.-H. Guo and N. J. Higham, *A Schur–Newton method for the matrix pth root and its inverse*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 788–804.
[7] J. M. Hawkins, *McMullen's root-finding algorithm for cubic polynomials*, Proc. Amer. Math. Soc., 130 (2002), pp. 2583–2592.
[8] N. J. Higham, *Stable iterations for the matrix square root*, Numer. Algorithms, 15 (1997), pp. 227–242.
[9] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008.
[10] N. J. Higham, D. S. Mackey, N. Mackey, and F. Tisseur, *Computing the polar decomposition and the matrix sign decomposition in matrix groups*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1178–1192.
[11] N. J. Higham, D. S. Mackey, N. Mackey, and F. Tisseur, *Functions preserving matrix groups and iterations for the matrix square root*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 849–877.
[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1990.
[13] A. S. Householder, *Principles of Numerical Analysis*, McGraw-Hill, New York, Toronto, London, 1953.
[14] J. L. Howland, *The sign matrix and the separation of matrix eigenvalues*, Linear Algebra Appl., 49 (1983), pp. 221–232.
[15] B. Iannazzo, *On the Newton method for the matrix pth root*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 503–523.
[16] Y. Jin and B. Kalantari, *Symmetric functions and root-finding algorithms*, Adv. in Appl. Math., 34 (2005), pp. 156–174.
[17] C. Kenney and A. J. Laub, *Rational iterative methods for the matrix sign function*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 273–291.
[18] S. Lakić, *On the computation of the matrix kth root*, ZAMM Z. Angew. Math. Mech., 78 (1998), pp. 167–172.
[19] C. McMullen, *Families of rational maps and iterative root-finding algorithms*, Ann. of Math. (2), 125 (1987), pp. 467–493.
[20] E. Schröder, *Über unendlich viele Algorithmen zur Auflösung der Gleichungen*, Math. Ann., 2 (1870), pp. 317–365.
[21] E. Schröder, *On Infinitely Many Algorithms for Solving Equations*, Technical report TR-92-121, University of Maryland, College Park, MD, 1992. Translated by G. W. Stewart.
[22] M. I. Smith, *A Schur algorithm for computing matrix pth roots*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 971–989.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1463–1482

THE QUADRATIC ARNOLDI METHOD FOR THE SOLUTION OF THE QUADRATIC EIGENVALUE PROBLEM*

KARL MEERBERGEN†

Abstract. The quadratic Arnoldi algorithm is a Krylov method for the solution of the quadratic eigenvalue problem, that exploits the structure of the Krylov vectors. This allows us to reduce the memory requirements by about a half. The method is an alternative to the second order Arnoldi (SOAR) method. In the SOAR method it is not clear how to perform an implicit restart. We discuss various choices of linearizations in $\mathbb{L}_1$ and $\mathbb{DL}$. We also explain how to compute a partial Schur form of the underlying linearization with respect to the structure of the Schur vectors. We also formulate some open problems.

Key words. quadratic eigenvalue problem, Arnoldi method, SOAR method, Schur decomposition

AMS subject classifications. 15A18, 65F15, 65F50

DOI. 10.1137/07069273X

1. Introduction.

The goal is to solve the quadratic eigenvalue problem

$$Q(\lambda)u = 0 \quad \text{with} \quad Q(\lambda) = K + \lambda C + \lambda^2 M. \tag{1.1}$$

The matrices K, $-iC$, and $-M$ are the stiffness, damping, and mass matrices, respectively, and arise from the Fourier transformation of the spatial discretization by finite elements of the equation of motion. The matrices are large $n \times n$ matrices and usually sparse. Equation (1.1) is solved when the engineer is interested in the eigenfrequencies (resonance frequencies) and damping properties of the mode shapes (i.e., the eigenvectors). Krylov methods for the solution of quadratic eigenvalue problems have been studied by Parlett and Chen [20], Saad [21], and Mehrmann and Watkins [18]. The quadratic eigenvalue problem and solution methods are reviewed by Tisseur and Meerbergen [25].

Standard methods cannot be used directly to efficiently solve (1.1) because of the quadratic term in λ. Instead, (1.1) can be "linearized" into a problem of the form

$$(A - \lambda B) \begin{pmatrix} \lambda u \\ u \end{pmatrix} = 0, \tag{1.2}$$

where A and B will be defined in section 2. Since we are interested in the eigenvalues near zero, we usually solve the inverted problem

$$S \begin{pmatrix} u \\ \theta u \end{pmatrix} = \theta \begin{pmatrix} u \\ \theta u \end{pmatrix} \tag{1.3}$$

with $S = A^{-1}B$ and $\theta = \lambda^{-1}$. There are three disadvantages to the Arnoldi method: first, the doubling of the size of the problem increases the memory cost by a factor of

*Received by the editors May 23, 2007; accepted for publication (in revised form) by J. H. Brandts August 11, 2008; published electronically December 3, 2008. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s).
http://www.siam.org/journals/simax/30-4/69273.html
†Department of Computer Science, K. U. Leuven, 3001 Heverlee (Leuven), Belgium (karl.meerbergen@cs.kuleuven.be).

two; second, the typical structure of the eigenvectors of the linearization is lost; and finally, the Ritz values are computed from a small Hessenberg matrix and not from a small quadratic eigenvalue problem.

All these disadvantages can be overcome by the second order Arnoldi (SOAR) method [4]. However, the SOAR method is not the preferred method for computing a Schur form of the linearization. As a consequence, classical implicit restarting [22, 19] is no longer possible.

In this document, we propose a method that is close to the SOAR method. We call it Q-Arnoldi, where Q stands for "quadratic." It exploits the structure of the linear problem to reduce the storage cost of the Arnoldi method roughly by a factor of two. We propose a locking procedure of converged Schur vectors that do keep the structure of the exact Schur vectors.

Both the SOAR method [4] and the Q-Arnoldi method compute the same subspace of the linearized problem. Q-Arnoldi performs a projection that does not produce a quadratic eigenvalue problem: the Ritz pairs are computed from the Arnoldi recurrence relation, which allows for the computation of a Schur form. Exploiting the Schur form of the linearization in the SOAR method for restarting purposes is not trivial and is, to date, an open question. See [15] for exploiting the Schur form in the inverse residual iteration and Jacobi–Davidson methods. The difficulty is that the Schur vectors, computed by the SOAR method, cannot, in general, be mapped in a Krylov recurrence relation.

Nevertheless, some results in this paper also have consequences concerning the SOAR method. We devote some time to the choice of linearization in $\mathbb{L}_1$ and $\mathbb{DL}$ [13], which is also useful for the SOAR method.

Note that $B^{-1}A$ can be used as an alternative to $A^{-1}B$. In applications this is usually less effective than using $A^{-1}B$ when the eigenvalues near zero are wanted. Many applications are not extremely large, i.e., smaller than 100,000 degrees of freedom, which allows the use of a direct linear system solver for applying A^{-1}.

The fact that the Krylov vectors belong to $\mathbb{C}^{2n}$ instead of $\mathbb{C}^n$ may limit their practical use, especially when the Arnoldi method [1, 12, 3] is used when a large number of vectors need to be stored: k iterations of the Arnoldi method require the storage of the order of $(2k+2)n$ floating point numbers (real or complex). The Q-Arnoldi scheme exploits the structure of S to reduce the memory requirements to $(k+2)n$ floating point numbers. Similar tricks can be used for reducing the storage cost for partial reorthogonalization in the Lanczos method. Although the Lanczos method [9, 10] can be used to keep the storage requirements low, the Arnoldi method is usually preferred for eigenvalue computations. This memory reduction trick is also used in the SOAR algorithm.

The eigenvectors of (1.1) appear twice in the eigenvectors of (1.2). When the linearized problems are solved in a Krylov subspace, however, the two computed solutions are, in general, different. Therefore, we address which one of the two should be returned as an approximate solution vector of (1.1).

The paper is organized as follows. In section 2, we introduce linearizations for (1.1). In section 3, we review the Arnoldi method for (1.2) and present a modification of Arnoldi's method that saves memory. We call this the Q-Arnoldi algorithm, where Q stands for "quadratic." We discuss various issues including the choice of linearization, some thoughts on computations in finite precision arithmetic, and the choice of component of the Ritz vectors. Section 4 shows how to exploit the structure of (1.2) in the computation of Schur vectors. In section 5 we show a numerical example from applications. We conclude with some remarks in section 6.

Throughout the paper, we use x^* to denote the Hermitian transpose, x^*y for the inner product of two vectors, and $\|x\| = \sqrt{x^*x}$ for the induced two-norm. We also use $\|\cdot\|$ to denote the two-norm of a matrix. The matrix Frobenius norm is denoted by $\|\cdot\|_F$.

2. Linearization via the first companion form. By linearization, we mean the transformation of (1.1) into (1.2) by a suitable choice of A and B so that there is a one-to-one correspondence between eigenpairs of (1.1) and the $A - \lambda B$ pencil [24, 14, 13]. The linearization should be chosen so that the application of $S = A^{-1}B$ is efficient and accurate. We therefore assume that A and so K are invertible. The motivation for using linearizations other than the classical companion forms lies in respecting the structure of the eigenvalue problem. In this section, we show that this structure is lost in the shift-and-invert Arnoldi method for the linearizations from [14, 13], meaning that the structure is not respected by the Arnoldi method.

The fact that K has no factor λ in (1.1) suggests that K should appear in A. A straightforward choice is

$$A = \begin{bmatrix} D & \\ & K \end{bmatrix}, \quad B = \begin{bmatrix} & D \\ -M & -C \end{bmatrix}, \quad y = \begin{pmatrix} \lambda x \\ x \end{pmatrix}, \tag{2.1}$$

where D can be any nonsingular matrix. It is easy to see that

$$S = A^{-1}B = \begin{bmatrix} & I \\ -K^{-1}M & -K^{-1}C \end{bmatrix}, \tag{2.2}$$

from which D disappears.

LEMMA 2.1. *The pencil $A-\lambda B$ with* (2.1) *is a linearization iff D is nonsingular.*

Proof. If D is nonsingular, $A - \lambda B$ is a linearization. If D were singular, $\lambda = 0$ is an eigenvalue of $A - \lambda B$ but not of (1.1). Also, $\lambda = \infty$ is an eigenvalue even if it is not an eigenvalue of (1.1). In addition, it is no longer guaranteed that all eigenvectors have the form (1.2), so $A - \lambda B$ is not a linearization. ∎

An alternative to (2.1) is the first companion form

$$B = \begin{bmatrix} -M & \\ & D \end{bmatrix}, \quad A = \begin{bmatrix} C & K \\ D & \end{bmatrix}, \quad y = \begin{pmatrix} \lambda x \\ x \end{pmatrix}, \tag{2.3}$$

which also produces (2.2).

The matrix used in [18] for the skew-Hamiltonian/Hamiltonian eigenvalue problem does not have the form (2.2).

The linearization can be chosen so that A and B respect the special structures of K, C, and M. For example, if all matrices are symmetric and M is nonsingular, one could use $D = -M$. Although $A^{-1}B$ is independent of D, the choice of D may help build the Krylov subspace more efficiently. We discuss this in more detail in section 3.2.

3. The Arnoldi method. Let $N = 2n$. The Arnoldi method applied to $S \in \mathbb{C}^{N\times N}$ and $b \in \mathbb{C}^N$ produces the Krylov subspace

$$\mathcal{K}_k(S, b) = \operatorname{span}\{b, Sb, S^2b, \ldots, S^{k-1}b\}.$$

It computes the $N \times k$ matrix $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$ of iteration vectors, the upper Hessenberg matrix H_k, and the residual term $\mathbf{v}_{k+1}\beta_k$ so that

$$\begin{aligned} S\mathbf{V}_k - \mathbf{V}_k H_k &= \mathbf{v}_{k+1}\beta_k e_k^*, \\ S\mathbf{V}_k - \mathbf{V}_{k+1}\underline{H}_k &= 0, \end{aligned} \tag{3.1}$$

where $\mathbf{V}_{k+1}^*\mathbf{V}_{k+1} = I$. Equation (3.1) is called the Arnoldi recurrence relation. An algorithm for computing $\mathbf{V}_k$ and H_k is now given.

ALGORITHM 3.1 (Arnoldi method).

1. *Set the initial vector* $\mathbf{v}_1$ *so that* $\|\mathbf{v}_1\|_2 = 1$.
2. *For* $j = 1, \ldots, k$ *do*
   - 2.1. *Compute* $\hat{\mathbf{v}}_j = S\mathbf{v}_j$.
   - 2.2. *Compute the Arnoldi coefficients* $h_j = \mathbf{V}_j^*\hat{\mathbf{v}}_j$.
   - 2.3. *Update* $\tilde{\mathbf{v}}_j = \hat{\mathbf{v}}_j - \mathbf{V}_j h_j$.
   - 2.4. *Get the scalar* $\beta_j = \|\tilde{\mathbf{v}}_j\|_2$ *and compute* $\mathbf{v}_{j+1} = \tilde{\mathbf{v}}_j/\beta_j$.

   End for

Steps 2.2–2.4 orthonormalize $S\mathbf{v}_j$ against $\mathbf{v}_1, \ldots, \mathbf{v}_j$ into $\mathbf{v}_{j+1}$. The coefficients h_j form the jth column of H_j, and β_j is the $(j+1, j)$th element of $\underline{H}_k$. Roughly speaking, k iterations cost about $2N(k+3)k$ flops, excluding the cost for Step 2.1, where one flop is the cost for an addition or a multiplication.

3.1. The Q-Arnoldi algorithm. We now discuss how we can make Algorithm 3.1 more efficient for S from (2.2). We decompose the jth Arnoldi vectors into

$$\mathbf{v}_j = \begin{pmatrix} v_j \\ w_j \end{pmatrix},$$

with $v_j, w_j \in \mathbb{C}^n$. The Arnoldi recurrence relation (3.1) for the linearization (2.1) can now be written as

$$\begin{bmatrix} I \\ -K^{-1}M & -K^{-1}C \end{bmatrix} \begin{pmatrix} V_k \\ W_k \end{pmatrix} - \begin{pmatrix} V_k \\ W_k \end{pmatrix} H_k = \beta_k \begin{pmatrix} v_{k+1} \\ w_{k+1} \end{pmatrix} e_k^*, \tag{3.2}$$

from which we deduce that

$$W_k = V_{k+1}\underline{H}_k. \tag{3.3}$$

This implies that we have only to store the vectors V_k, v_{k+1}, and w_{k+1} to evaluate the recurrence relation, which contain $(2+k)n$ floating point numbers. Storing only V_k, v_{k+1}, and w_{k+1} results in an important reduction of the memory cost compared to Algorithm 3.1. The following algorithm implements this idea.

ALGORITHM 3.2 (Q-Arnoldi).

1. *Let* v_1 *and* w_1 *be chosen so that* $\|v_1\|_2^2 + \|w_1\|_2^2 = 1$.
2. *For* $j = 1, \ldots, k$ *do*
   - 2.1. *Compute* $\hat{w}_j = -K^{-1}(Mv_j + Cw_j)$ *and* $\hat{v}_j = w_j$.
   - 2.2. *Compute the Arnoldi coefficients*

$$h_j = \begin{bmatrix} V_{j-1}^*\hat{v}_j + \underline{H}_{j-1}^*(V_j^*\hat{w}_j) \\ v_j^*\hat{v}_j + w_j^*\hat{w}_j \end{bmatrix}.$$

   - 2.3. *Update*

$$\tilde{v}_j = \hat{v}_j - V_j h_j,$$
$$\tilde{w}_j = \hat{w}_j - [V_j \;\; w_j]\left(\begin{bmatrix} \underline{H}_{j-1} & 0 \\ 0 & 1 \end{bmatrix} h_j\right).$$

   - 2.4. *Normalize* $v_{j+1} = \tilde{v}_j/\beta_j$ *and* $w_{j+1} = \tilde{w}_j/\beta_j$ *with* $\beta_j = (\|\tilde{v}_j\|^2 + \|\tilde{w}_j\|^2)^{1/2}$.
   - 2.5. *Set the* j*th column of* $\underline{H}_j$ *as* $[h_j^T \;\; \beta_j]^T$.

   End for

The difference between Algorithm 3.1 and Algorithm 3.2 is only in Steps 2.2–2.3, where W_{j-1} is replaced by $V_j \underline{H}_{j-1}$. The cost for computing $\underline{H}_{j-1}^*(V_j^* \hat{w}_j)$ in Step 2.2 is $2(nj+j(j-1))$ flops, so that the total cost for Step 2.2 is $2(2nj+n+j(j-1))$ flops. The cost for computing $\tilde{w}_j$ in Step 2.3 is $2((j-1)j+n(j+1))$, and the computation of $\tilde{v}_j$ requires $2nj$ flops. Step 2.4 costs $8n$ flops, as for Algorithm 3.1. The total cost for Steps 2.2–2.4 is $8nj + 12n + 4j(j-1)$. For k iterations the cost is of the order of $4nk^2 + 16nk + \frac{4}{3}(k-1)k(k+1)$ flops. Algorithm 3.1 requires $4n(k+1)k$ flops. So, when k is significantly smaller than n (which is usually the case; otherwise the Arnoldi method is not suitable anyway), Steps 2.2–2.4 of Algorithms 3.1 and 3.2 have a cost of approximately $4nk^2$. Note that only $v_1, \ldots, v_{j+1}$ and w_{j+1} need to be stored on iteration j so that the memory requirements for the storage of the Arnoldi vectors for k iterations is limited to $n(k+2)$ with the Q-Arnoldi method. The storage for the Arnoldi method is of the order of $2n(k+1)$. Although Q-Arnoldi is slightly more expensive in computation time, the extra cost is usually small compared to the computation of $A^{-1}B\mathbf{v}_j$.

3.2. Other linearizations. In this section, we study the use of linearizations other than the companion form in (2.3). These are introduced for respecting special structures in the eigenvalues. Consider the vectorspace of linearizations [14] of the form

$$\mathbb{L}_1(Q) = \left\{ A - \lambda B : (A - \lambda B) \begin{pmatrix} \lambda \\ 1 \end{pmatrix} = \begin{pmatrix} \eta_1 Q(\lambda) \\ \eta_2 Q(\lambda) \end{pmatrix}, \eta_{1,2} \in \mathbb{C} \right\}, \tag{3.4}$$

where Q is defined in (1.1). From (3.4), we have

$$A = \begin{bmatrix} A_{11} & \eta_1 K \\ A_{21} & \eta_2 K \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} -\eta_1 M & A_{11} - \eta_1 C \\ -\eta_2 M & A_{21} - \eta_2 C \end{bmatrix},$$

where A_{11}, A_{21}, η_1, and η_2 can be freely chosen.

THEOREM 3.1. *Let A and B be defined following* (3.4)*, and let K be invertible. Then $A - \lambda B$ is a linearization of* (1.1) *iff A is invertible. In addition,* (2.2) *holds, and applying the Arnoldi method to $A^{-1}B$ produces Arnoldi vectors with the structure* (3.3).

Proof. Let

$$S = A^{-1}B = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}.$$

From $B = AS$, i.e.,

$$\begin{bmatrix} -\eta_1 M & A_{11} - \eta_1 C \\ -\eta_2 M & A_{21} - \eta_2 C \end{bmatrix} = \begin{bmatrix} A_{11} & \eta_1 K \\ A_{21} & \eta_2 K \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix},$$

it is easy to verify (2.2).

Formally, A does not need to be invertible for $AS = B$ to be true. If both $\eta_1 = \eta_2 = 0$, then $\det(A - \lambda B) = 0$ for all λ's, so $A - \lambda B$ is not a linearization. Suppose $\eta_1 \neq 0$. If we multiply the first block of A and B by η_2 and the second block row by η_1, we find the pencil $\tilde{A} - \tilde{B}$ with the same eigenvalues and eigenvectors, where

$$\tilde{A} = \begin{bmatrix} A_{11} & \eta_1 K \\ \eta_2 A_{11} - \eta_1 A_{21} & 0 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} -\eta_1 M & A_{11} - \eta_1 C \\ 0 & \eta_2 A_{11} - \eta_1 A_{21} \end{bmatrix}.$$

Since K is nonsingular, A can be singular only when

$$\eta_2 A_{11} - \eta_1 A_{21}$$

is singular. Following Lemma 2.1, the pencil $\tilde{A} - \tilde{B}$ can only be a linearization when $\eta_2 A_{11} - \eta_1 A_{21}$ is nonsingular. In other words, A should be invertible for $A - \lambda B$ being a linearization of (1.1).

Since $A^{-1}B$ is the same matrix for all linearizations of this form, the Arnoldi method produces the same recurrence relation, which finishes the proof. ∎

One example that fits into this framework is the solution of palindromic eigenvalue problems [13]. Let $K = M^T$ and $C = C^T$; then (1.1) is called T-palindromic. If (λ, x) is an eigenpair to (1.1) and the associated left eigenvector is y so that $y^* Q(\lambda) = 0$, then λ^{-1} also is an eigenvalue with (right) eigenvector $\bar{y}$ and left eigenvector $\bar{x}$. In [13], methods that produce eigenvalue approximates that respect this spectral structure are advocated. They therefore introduce the linearization with $B = -A^T$ in (3.4) of the form

$$A = \begin{bmatrix} K & K \\ C - M & K \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} -M & K - C \\ -M & -M \end{bmatrix}.$$

This corresponds to $\eta_1 = \eta_2 = 1$, $A_{11} = K$, and $A_{21} = C - M$. A is invertible when K and $C - M - K$ are invertible, i.e., 0 and -1 are not eigenvalues of (1.1). The linearized pencil $A + \lambda(-B)$ is T-palindromic, since $(-B) = A^T$. Using $A^{-1}B$ in the Arnoldi method requires the inverse of K and $C - M - K = C - K - K^T$. However, note that the Ritz values from Arnoldi's method do not necessarily come in pairs of the form λ, λ^{-1}.

The generalization of the second companion form is given by

$$\mathbb{L}_2(Q) = \{A - \lambda B : (\lambda I \quad I)(A - \lambda B) = (\tilde{\eta}_1 Q(\lambda) \quad \tilde{\eta}_2 Q(\lambda)), \tilde{\eta}_{1,2} \in \mathbb{C}\}.$$

Similarly to $\mathbb{L}_1$, we can show that A and B take the form

$$A = \begin{bmatrix} A_{11} & A_{12} \\ \tilde{\eta}_1 K & \tilde{\eta}_2 K \end{bmatrix} \quad \text{with} \quad B = \begin{bmatrix} -\tilde{\eta}_1 M & -\tilde{\eta}_2 M \\ A_{11} - \tilde{\eta}_1 C & A_{12} - \tilde{\eta}_2 C \end{bmatrix}$$

and

$$BA^{-1} = \begin{bmatrix} 0 & -MK^{-1} \\ I & -CK^{-1} \end{bmatrix}.$$

Note that $A^{-1}B$ depends on A_{11}, A_{12}, and $\tilde{\eta}_{1,2}$.

The intersection of $\mathbb{L}_1$ and $\mathbb{L}_2$ is denoted by $\mathbb{DL}(Q)$ [14]. Its general form is

$$A = \begin{bmatrix} \eta_1 C - \eta_2 M & \eta_1 K \\ \eta_1 K & \eta_2 K \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} -\eta_1 M & -\eta_2 M \\ -\eta_2 M & -\eta_2 C + \eta_1 K \end{bmatrix}.$$

The common cases for symmetric K, C, and M are $\eta_{1,2} = \{0, 1\}$ and $\eta_{1,2} = \{1, 0\}$. Note that A is invertible iff

$$\eta_1 \eta_2 C - \eta_2^2 M - \eta_1^2 K$$

is invertible, i.e., $-\eta_2/\eta_1$ is not an eigenvalue of (1.1). Although working with pencils in $\mathbb{DL}$ does not seem important for the Arnoldi method, it might be of interest when using the Lanczos method; see [20, 2]. The pseudo-Lanczos method [20] is the Lanczos method applied to $A^{-1}B$ using the B pseudo-inner product; i.e., the Lanczos vectors are orthogonal with respect to B, where A and B are chosen in $\mathbb{DL}$. Using the B inner product $x^T B y$ rather than the Euclidean inner product $x^* y$ produces a tridiagonal H_k.

3.3. Applying a shift. In this section, we review ideas for shifting the eigenvalue problem, which is a key element in the shift-and-invert Arnoldi method. We first apply a shift to (1.1) in section 3.3.1, and then to the linearization (2.1) in section 3.3.2.

3.3.1. Shifting the quadratic equation. The convergence of eigenvalues near σ can be improved [21, 3] by shifting the eigenvalue problem (1.1) into

$$(\tilde{K} + (\lambda - \sigma)\tilde{C} + (\lambda - \sigma)^2\tilde{M})u = 0, \tag{3.5}$$

with

$$\tilde{K} = K + \sigma C + \sigma^2 M, \quad \tilde{C} = C + 2\sigma M, \quad \tilde{M} = M.$$

Without loss of generality, we can assume that $\sigma = 0$, by replacing K, C, and M by $\tilde{K}$, $\tilde{C}$, and $\tilde{M}$, respectively, and $\lambda - \sigma$ by λ.

3.3.2. Shifting the linearization. We can also shift (2.1) into

$$(A - \sigma B)^{-1}(A - \lambda B)y(\lambda) = 0.$$

The recurrence relation for $(A - \sigma B)^{-1}B$ becomes

$$\begin{bmatrix} A_{11} + \sigma\eta_1 M & \eta_1 K - \sigma A_{11} + \sigma\eta_1 C \\ A_{21} + \sigma\eta_2 M & \eta_2 K - \sigma A_{21} + \sigma\eta_2 C \end{bmatrix}^{-1} \begin{bmatrix} -\eta_1 M & A_{11} - \eta_1 C \\ -\eta_2 M & A_{21} - \eta_2 C \end{bmatrix} \begin{pmatrix} V_k \\ W_k \end{pmatrix}$$

$$= \begin{pmatrix} V_{k+1} \\ W_{k+1} \end{pmatrix} \underline{H}_k.$$

With $\tilde{A}_{j1} = A_{j1} + \sigma\eta_j M$ this becomes

$$\begin{bmatrix} \tilde{A}_{11} & \eta_1\tilde{K} - \sigma\tilde{A}_{11} \\ \tilde{A}_{21} & \eta_2\tilde{K} - \sigma\tilde{A}_{21} \end{bmatrix}^{-1} \begin{bmatrix} -\eta_1 M & \tilde{A}_{11} - \eta_1\tilde{C} + \eta_1\sigma M \\ -\eta_2 M & \tilde{A}_{21} - \eta_2\tilde{C} + \eta_2\sigma M \end{bmatrix} \begin{pmatrix} V_k \\ W_k \end{pmatrix}$$

$$= \begin{pmatrix} V_{k+1} \\ W_{k+1} \end{pmatrix} \underline{H}_k.$$

The pencil $(A - \sigma B) - \mu B$ does not lie in $\mathbb{L}_1(Q(\lambda - \sigma))$. However, when we introduce $Z_{k+1} = V_{k+1} - \sigma W_{k+1}$, we have

$$\begin{bmatrix} \tilde{A}_{11} & \eta_1\tilde{K} \\ \tilde{A}_{21} & \eta_2\tilde{K} \end{bmatrix}^{-1} \begin{bmatrix} -\eta_1 M & \tilde{A}_{11} - \eta_1\tilde{C} \\ -\eta_2 M & \tilde{A}_{21} - \eta_2\tilde{C} \end{bmatrix} \begin{pmatrix} Z_k \\ W_k \end{pmatrix} = \begin{pmatrix} Z_{k+1} \\ W_{k+1} \end{pmatrix} \underline{H}_k,$$

which is exactly the same as (3.2) with $V_{k+1} = Z_{k+1}$, where $\tilde{K}$, $\tilde{C}$, and M are related to the shifted problem (3.5). We compute $W_k = Z_{k+1}\underline{H}_k$ and $V_k = Z_{k+1} + \sigma W_k$.

3.4. Numerical stability. A few words about numerical stability are in order. In this section, we perform a traditional rounding error analysis on Algorithms 3.1 and 3.2.

The fact that we compute W_k from (3.3) changes the Arnoldi method in finite precision arithmetic. In this section, we show that, under certain conditions, the

Q-Arnoldi method is as backward stable for the recurrence relation as the Arnoldi method. More specifically, we shall give bounds to

$$\tau_R = \|W_k - V_{k+1}\underline{H}_k\|_F, \tag{3.6}$$

where $\mathbf{V}_{k+1}$ and $\underline{H}_k$ are computed by either the Arnoldi method or the Q-Arnoldi method.

The reader is referred to Higham [8, Chapter 3] for more details on computations with finite precision arithmetic. We denote by u the machine precision. We introduce the symbol $\lesssim$ to denote

$$\alpha \lesssim \beta \Leftrightarrow \alpha \leq c\beta + \mathcal{O}(u),$$

where c is a u independent constant. We also bound $\||X|\cdot|Y|\|_F \leq \|X\|_F \cdot \|Y\|_F$.

We define the error matrices F_j and G_j on the recurrence relation (3.1):

$$\hat{V}_j - V_{j+1}\underline{H}_j = F_j, \tag{3.7a}$$

$$\hat{W}_j - W_{j+1}\underline{H}_j = G_j. \tag{3.7b}$$

3.5. Analysis for the Arnoldi algorithm. We first show two lemmas that will help us to make a statement on the backward stability of the Arnoldi method.

LEMMA 3.2. *For all $j = 1, \ldots, k$ we have*

$$\|v_{j+1}\beta_j - \tilde{v}_j\| \lesssim u\|\tilde{v}_j\| \lesssim u\|v_{j+1}\|\beta_j, \tag{3.8a}$$

$$\|w_{j+1}\beta_j - \tilde{w}_j\| \lesssim u\|\tilde{w}_j\| \lesssim u\|w_{j+1}\|\beta_j, \tag{3.8b}$$

$$\left\|\begin{pmatrix} v_{j+1} \\ w_{j+1}\end{pmatrix}\beta_j - \begin{pmatrix} \tilde{v}_j \\ \tilde{w}_j\end{pmatrix}\right\| \lesssim u\left\|\begin{pmatrix} \tilde{v}_j \\ \tilde{w}_j\end{pmatrix}\right\| \lesssim u\beta_j. \tag{3.8c}$$

We also have

$$\|V_j\|_F^2 + \|W_j\|_F^2 = j + \mathcal{O}(u). \tag{3.9}$$

Proof. The proofs of (3.8a), (3.8b), and (3.8c) follow from the fact that v_{j+1} and w_{j+1} are computed as $v_{j+1} = \tilde{v}_j/\beta_j$ and $w_{j+1} = \tilde{w}_j/\beta_j$. The proof of (3.9) follows from [8, Theorem 18.2] with $m = 1$ for $\|v_j\|^2 + \|w_j\|^2 = 1 + \mathcal{O}(u)$. ∎

LEMMA 3.3. *In finite precision arithmetic, Algorithm* 3.1 *produces V_{j+1}, W_{j+1}, and $\underline{H}_j$ so that* (3.7) *holds with*

$$\|F_j\|_F \lesssim u\||V_{j+1}|\cdot|\underline{H}_j|\|_F \lesssim u\|V_{j+1}\|_F\|\underline{H}_j\|_F, \tag{3.10a}$$

$$\|G_j\|_F \lesssim u\||W_{j+1}|\cdot|\underline{H}_j|\|_F \lesssim u\|W_{j+1}\|_F\|\underline{H}_j\|_F, \tag{3.10b}$$

$$\left\|\begin{pmatrix} \hat{V}_j - V_{j+1}\underline{H}_j \\ \hat{W}_j - W_{j+1}\underline{H}_j\end{pmatrix}\right\|_F \lesssim u\|\underline{H}_j\|_F. \tag{3.10c}$$

In addition, we have

$$\left\|h_j - \begin{pmatrix} V_j \\ W_j\end{pmatrix}^* \begin{pmatrix} \hat{v}_j \\ \hat{w}_j\end{pmatrix}\right\| \lesssim u\left\|\begin{pmatrix} \hat{v}_j \\ \hat{w}_j\end{pmatrix}\right\|. \tag{3.11}$$

Proof. Recall that we can omit constant factors from error bounds using the notation $\lesssim$. We first prove (3.10a). We define

$$\tilde{f}_j = \hat{v}_j - V_j h_j - \tilde{v}_j, \tag{3.12}$$

where $\tilde{v}_j$ is computed in Step 2.3 of Algorithm 3.1. We have

$$\|\tilde{f}_j\| \lesssim u(\|\hat{v}_j\| + \||V_j| \cdot |h_j|\|).$$

Denote by f_j the last column of F_j:

$$f_j = \hat{v}_j - V_j h_j - v_{j+1}\beta_j = \tilde{f}_j + (\tilde{v}_j - v_{j+1}\beta_j), \tag{3.13}$$

where v_{j+1} and β_j are computed from Step 2.4. The application of Lemma 3.2 on Step 2.4 of Algorithm 3.1 readily produces

$$\|f_j\| \lesssim u(\|\hat{v}_j\| + \||V_j| \cdot |h_j|\| + \|v_{j+1}\|\beta_j) \lesssim u \left\| |V_{j+1}| \cdot \left| \begin{pmatrix} h_j \\ \beta_j \end{pmatrix} \right| \right\|,$$

using $\|\hat{v}_j\| \lesssim \||V_j||h_j|+|v_{j+1}|\beta_j\|$. Accumulating f_j in $F_j = [f_1, \ldots, f_j]$ proves (3.10a). The proofs for (3.10b) and (3.10c) are similar. We use (3.9) to bound (3.10c).

The proof of (3.11) readily follows from standard rounding error analysis. ∎

3.6. Analysis for the Q-Arnoldi algorithm. We define

$$\delta_j = \|V_j\|_F, \tag{3.14}$$

$$\gamma_j = \|[|V_j| \cdot |\underline{H}_{j-1}| \quad w_j]\|_F, \tag{3.15}$$

$$\tilde{\gamma}_{j+1} = \|[|V_j| \cdot |\underline{H}_{j-1}| \quad w_j \quad w_{j+1}]\|_F. \tag{3.16}$$

For the Q-Arnoldi algorithm, $W_j z$ and $\begin{pmatrix} V_j \\ W_j \end{pmatrix}^* \begin{pmatrix} \hat{v}_j \\ \hat{w}_j \end{pmatrix}$ are computed as

$$[V_j \underline{H}_{j-1} \quad w_j]z \quad \text{and} \quad \begin{pmatrix} \underline{H}_{j-1}^*(V_j^* \hat{w}_j) \\ w_j^* \hat{w}_j \end{pmatrix} + V_j^* \hat{v}_j, \tag{3.17}$$

respectively. The componentwise errors on the computation of (3.17) are of the order

$$u[|V_j| \cdot |\underline{H}_{j-1}| \quad |w_j|]|z| \quad \text{and} \quad u \left| \begin{pmatrix} |\underline{H}_{j-1}|^*(|V_j|^*|\hat{w}_j|) \\ |w_j|^*|\hat{w}_j| \end{pmatrix} + |V_j|^*|\hat{v}_j| \right|, \tag{3.18}$$

respectively. The normwise error bounds are

$$u\gamma_j\|z\| \quad \text{and} \quad u(\gamma_j + \delta_j) \left\| \begin{pmatrix} \hat{v}_j \\ \hat{w}_j \end{pmatrix} \right\|, \tag{3.19}$$

respectively.

We extend Lemma 3.3 to the Q-Arnoldi algorithm.

LEMMA 3.4. *In finite precision arithmetic, Algorithm* 3.2 *produces* V_{j+1}, W_{j+1}, *and* $\underline{H}_j$ *so that* (3.7) *holds with*

$$\|F_j\|_F \lesssim u\delta_{j+1}\|\underline{H}_j\|_F, \tag{3.20a}$$

$$\|G_j\|_F \lesssim u\gamma_{j+1}\|\underline{H}_j\|_F, \tag{3.20b}$$

$$\left\| \begin{pmatrix} \hat{V}_j - V_{j+1}\underline{H}_j \\ \hat{W}_j - W_{j+1}\underline{H}_j \end{pmatrix} \right\|_F \lesssim u\sqrt{\gamma_{j+1}^2 + \delta_{j+1}^2}\|\underline{H}_j\|_F. \tag{3.20c}$$

Proof. Similar to the proof of (3.10a), we find for the componentwise analysis that

$$|F_j| = |W_j - V_{j+1}\underline{H}_j| \lesssim u|V_{j+1}| \cdot |\underline{H}_j|,$$

which proves (3.20a).

We now prove (3.20b). We define

$$\tilde{g}_j = \hat{w}_j - [V_j \underline{H}_{j-1} \quad w_j] h_j - \tilde{w}_j, \tag{3.21}$$

where $\tilde{w}_j$ is computed in Step 2.3 of Algorithm 3.2. We have

$$\|\tilde{g}_j\| \lesssim u(\|\hat{w}_j\| + \left\| \left[\begin{array}{cc} |V_j| \cdot |\underline{H}_{j-1}| & |w_j| \end{array} \right] |h_j| \right\|).$$

Let

$$g_j = \hat{w}_j - W_j h_j - w_{j+1}\beta_j, \tag{3.22}$$

$$= \tilde{g}_j + (\tilde{w}_j - w_{j+1}\beta_j) + ([V_j \underline{H}_{j-1} \quad w_j] - W_j)h_j, \tag{3.23}$$

where w_{j+1} and β_j are computed from Step 2.4. Using Lemma 3.2, we find that

$$\begin{aligned} \|g_j\| &\lesssim u(\|\hat{w}_j\| + \|[|V_j| \cdot |\underline{H}_{j-1}| \quad |w_j|]|h_j|\| + \|w_{j+1}\|\beta_j + \|[|V_j| \cdot |\underline{H}_{j-1}| \quad 0]|h_j|\|) \\ &\lesssim u \left\| [|V_j| \cdot |\underline{H}_{j-1}| \quad |w_j| \quad |w_{j+1}|] \left| \begin{pmatrix} h_j \\ \beta_j \end{pmatrix} \right| \right\|. \end{aligned}$$

Accumulating g_j in $G_j = [g_1, \ldots, g_j]$ and noting that $\tilde{\gamma}_{j+1} \leq \gamma_{j+1}$ proves (3.20b). The proof for (3.20c) is similar. ∎

Let $\sigma_{\min}$ and $\sigma_{\max}$ denote the smallest and largest singular values, respectively. Define

$$\xi_{j,\min} = \sqrt{\sigma_{\min}\left(\underline{H}_{j-1}\right)^2 + 1}, \tag{3.24}$$

$$\xi_{j,\max} = \sqrt{\sigma_{\max}\left(\underline{H}_{j-1}\right)^2 + 1}. \tag{3.25}$$

Note that $\xi_{j,\min} \leq 1 \leq \xi_{j,\max}$. For vibration problems, S usually has small eigenvalues, usually leading to $\xi_{j,\min} \approx 1$. The choice of pole σ (see section 3.3) may influence the large singular values of S. It is common practice not to pick the pole close to an eigenvalue when several eigenvalues are wanted [7, 16]. In this case, $\xi_{j,\max}$ is not large.

THEOREM 3.5. *In exact arithmetic,*

$$\left\| \begin{pmatrix} V_{j-1} \\ |V_j| \cdot |\underline{H}_{j-1}| \end{pmatrix} \right\|_2 \leq \frac{\xi_{j,\max}}{\xi_{j,\min}}.$$

Proof. First,

$$\sigma_{\min}\left(\begin{pmatrix} I_{j-1} \\ \underline{H}_{j-1} \end{pmatrix} \right) = \lambda_{\min}^{1/2}(I + \underline{H}_{j-1}^* \underline{H}_{j-1}) = \sqrt{\sigma_{\min}\left(\underline{H}_{j-1}\right)^2 + 1} = \xi_{j,\min}.$$

Similarly,

$$\sigma_{\max}\left(\begin{pmatrix} I_{j-1} \\ \underline{H}_{j-1} \end{pmatrix} \right) = \xi_{j,\max}.$$

From

$$\begin{pmatrix} V_{j-1} \\ V_j \underline{H}_{j-1} \end{pmatrix} = \begin{bmatrix} V_j & 0 \\ 0 & V_j \end{bmatrix} \begin{pmatrix} \underline{I}_{j-1} \\ \underline{H}_{j-1} \end{pmatrix}$$

and

$$\left\| \begin{pmatrix} V_{j-1} \\ V_j \underline{H}_{j-1} \end{pmatrix} \right\|_2 = 1,$$

we have that

$$\xi_{j,\max}^{-1} \leq \|V_j\|_2 \leq \xi_{j,\min}^{-1}.$$

The proof follows from

$$\left\| \begin{bmatrix} V_j & 0 \\ 0 & V_j \end{bmatrix} \begin{pmatrix} \underline{I}_{j-1} \\ \underline{H}_{j-1} \end{pmatrix} \right\| \leq \xi_{j,\min}^{-1} \xi_{j,\max}. \qquad \square$$

The conclusion from this section is that a loss of precision is possible in the computation of $\underline{H}_k$ (see (3.11) and (3.19)) and the recurrence relation in the Q-Arnoldi process when $\xi_{k,\max}/\xi_{k,\min}$ is large.

4. The solution of the quadratic eigenvalue problem. To simplify, we write S, defined in (2.2), in the form

$$S = \begin{bmatrix} & I \\ S_1 & S_2 \end{bmatrix}.$$

The solution of the quadratic eigenvalue problem by the shift-and-invert Arnoldi method is the objective of this section. For the computation of a number of eigenvalues of a non-Hermitian linear problem, we usually compute a partial Schur form. The idea is that we want the computed Schur vectors to have the structure of the exact Schur vectors, i.e., the form

$$\begin{pmatrix} U_k \\ U_k S_k \end{pmatrix}, \tag{4.1}$$

where S_k is the (upper-triangular) Schur matrix. The Schur vectors from the Arnoldi method do not have the form (4.1). It turns out that when we force the Schur vectors to satisfy the structure (4.1), we can keep the structure of the Krylov vectors in the form (3.3). In addition, implicit restarting also maintains the structure of the Krylov vectors.

We first introduce the notion of the Q-Arnoldi triple.

4.1. Definition and properties of Q-Arnoldi triples.

DEFINITION 4.1. *$\mathcal{Q} = \{V_{k+1}, \underline{H}_k, w_{k+1}\}$ is a Q-Arnoldi triple associated with S iff $V_{k+1} \in \mathbb{C}^{n\times(k+1)}$, $w_{k+1} \in \mathbb{C}^n$, $\underline{H}_k \in \mathbb{C}^{(k+1)\times k}$, and for*

$$\mathbf{V}_k = \begin{pmatrix} V_k \\ V_{k+1}\underline{H}_k \end{pmatrix} \quad \text{and} \quad \mathbf{V}_{k+1} = \begin{pmatrix} V_k & v_{k+1} \\ V_{k+1}\underline{H}_k & w_{k+1} \end{pmatrix}, \tag{4.2}$$

1. *the Arnoldi recurrence relation* (3.1) *holds, and*
2. *the Arnoldi vectors are orthogonal:*

$$\|I - \mathbf{V}_{k+1}^* \mathbf{V}_{k+1}\| = 0. \tag{4.3}$$

DEFINITION 4.2. *We denote by $\mathbf{Q}_k(S)$ the set of all Q-Arnoldi triples associated with S.*

4.1.1. Inexact Q-Arnoldi triples. In practice, we may allow a small error on (3.1) and (4.3) so that

$$S\mathbf{V}_k - \mathbf{V}_{k+1}\underline{H}_k = F_k \tag{4.4}$$

and

$$\left\|I - \mathbf{V}_{k+1}^*\mathbf{V}_{k+1}\right\| = \gamma_k, \tag{4.5}$$

where F_k can be considered a backward error on S.

DEFINITION 4.3. *The set of inexact Q-Arnoldi triples*

$$\mathbf{Q}_k(S, \eta, \rho)$$

consists of $\{V_{k+1}, \underline{H}_k, w_{k+1}\}$ *that satisfy* (4.4), (4.5), *and* (4.2) *with* $\|F_k\| \leq \eta$ *and* $\gamma_k \leq \rho$.

4.1.2. Transformations of (inexact) Q-Arnoldi triples.

DEFINITION 4.4. *Let* Z *be a full column rank* $k+1 \times p+1$ *matrix of the form* $Z = \left(\begin{smallmatrix} Z_1 & z \\ 0 & \zeta \end{smallmatrix}\right)$ *with* Z_1 *a* $k \times p$ *matrix and* $k \geq p$. *We define the transformation*

$$\mathcal{T}_Z\ \{V_{k+1}, \underline{H}_k, w_{k+1}\} = \{V_{k+1}Z, Z^\dagger \underline{H}_k Z_1, V_{k+1}\underline{H}_k z + w_{k+1}\zeta\},$$

where $Z^\dagger$ *is the generalized inverse, i.e.,* $Z^\dagger Z = I$.

The following theorem characterizes the transformation of an (inexact) Q-Arnoldi triple.

THEOREM 4.5. *Let* $\mathcal{Q} \in \mathbf{Q}_k(S, \eta, \rho)$. *Let* $\mathcal{T}_Z$ *be a transformation as defined by Definition* 4.4. *If*

$$ZZ^\dagger \underline{H}_k Z_1 = \underline{H}_k Z_1,$$

then

$$\mathcal{T}_Z\mathcal{Q} \in \mathbf{Q}_p(S, \eta\|Z_1\|, \|Z\|^2\rho + \|I - Z^*Z\|).$$

Proof. Let $\mathcal{Q} = \{V_{k+1}, \underline{H}_k, w_{k+1}\} \in \mathbf{Q}_k(S, \eta, \rho)$. Under the condition of the theorem, the elements of $\mathcal{T}_Z\mathbf{Q}_k(S)$ respect the structure (4.2). Multiplication of (4.4) on the right by Z_1 proves that the error on the recurrence relation of the transformed triple is bounded by $\|Z_1\|\eta$. Finally, we have that

$$I - Z^*\mathbf{V}_{k+1}^*\mathbf{V}_{k+1}Z = Z^*(I - \mathbf{V}_{k+1}^*\mathbf{V}_{k+1})Z + (I - Z^*Z),$$

which proves the theorem. ∎

When Z is square, Theorem 4.5 always holds. When, in addition to the conditions of Theorem 4.5, $Z^*Z = I$, $\mathcal{T}_Z\mathbf{Q}_k(S) \subset \mathbf{Q}_p(S)$.

4.1.3. Modification of the vectors of a Q-Arnoldi triple. Let $\{V_{k+1}, \underline{H}_k, w_{k+1}\} \in \mathbf{Q}_k(S)$. Suppose we modify the second block of the Arnoldi vectors as follows: $\tilde{W}_k = W_k + v_{k+1}g^*$. Not surprisingly, (4.2), (3.1), and (4.3) are broken. With $\tilde{\mathbf{V}}_k = \left(\begin{smallmatrix} V_k \\ \tilde{W}_k \end{smallmatrix}\right)$, we have that the recurrence relation becomes

$$S\tilde{\mathbf{V}}_k - \tilde{\mathbf{V}}_{k+1}\underline{H}_k = G_k := \begin{pmatrix} v_{k+1}g^* \\ S_2 v_{k+1}g^* - v_{k+1}g^*H_k \end{pmatrix},$$

where

$$\|G_k\| \leq (\|S_2 v_{k+1}\| + \|v_{k+1}\|\|H_k\| + 1)\|g\|. \tag{4.6}$$

The structure of the vectors (4.2) can be restored by modifying $\underline{\tilde{H}}_k = \underline{H}_k + e_{k+1}g^*$. The recurrence relation for the new $\underline{H}_k$ becomes

$$S\tilde{\mathbf{V}}_k - \tilde{\mathbf{V}}_{k+1}\underline{\tilde{H}}_k = \tilde{G}_k := \begin{pmatrix} 0 \\ S_2 v_{k+1}g^* - v_{k+1}g^* H_k - w_{k+1}g^* \end{pmatrix}. \tag{4.7}$$

The upper bound (4.6) also holds for $\tilde{G}_k$. The orthogonality of $\tilde{V}_k$ is restored by applying $\mathcal{T}_{Z^{-1}}$ to the Q-Arnoldi triple, where Z is upper-triangular and so that $\tilde{\mathbf{V}}_k^* \tilde{\mathbf{V}}_k = Z^* Z$, which can be computed by a Cholesky factorization.

Suppose we modify the first block of the Arnoldi vectors as follows: $\hat{V}_k = V_k + v_{k+1}g^*$; then (4.2) is restored by modifying $\underline{\hat{H}}_k = \underline{H}_k - e_{k+1}g^* H_k$. With $\hat{\mathbf{V}}_k = \left(\begin{smallmatrix} \hat{V}_k \\ W_k \end{smallmatrix}\right)$, we have that the recurrence relation for the new vectors becomes

$$S\hat{\mathbf{V}}_k - \hat{\mathbf{V}}_{k+1}\underline{\hat{H}}_k = \begin{pmatrix} 0 \\ S_1 v_{k+1}g^* - w_{k+1}g^* H_k \end{pmatrix}. \tag{4.8}$$

The orthogonality can be restored in the same way as for $\tilde{\mathbf{V}}_k$.

4.2. Computing Ritz vectors and Schur vectors.

4.2.1. Ritz vectors. The Ritz vectors corresponding to Ritz value θ have the form $x = \left(\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right) = \left(\begin{smallmatrix} V_k z \\ V_{k+1}\underline{H}_k z \end{smallmatrix}\right)$, where $H_k z = \theta z$. When (θ, x) is an eigenpair of S, $x_2 = \theta x_1$. As a Ritz vector of (1.1), we can return x_2/θ or x_1.

In this section, we study which of these is the best choice. Let the residual of the Ritz pair computed by the Arnoldi method be

$$\begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{bmatrix} 0 & I \\ S_1 & S_2 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \theta \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_2 - \theta x_1 \\ S_1 x_1 + S_2 x_2 - \theta x_2 \end{pmatrix}. \tag{4.9}$$

If we use x_1 as a Ritz vector for (1.1), the Ritz vector for the linearization is then

$$\tilde{x} = \begin{pmatrix} x_1 \\ \theta x_1 \end{pmatrix}. \tag{4.10}$$

The residual is

$$\begin{bmatrix} 0 & I \\ S_1 & S_2 \end{bmatrix} \begin{pmatrix} x_1 \\ \theta x_1 \end{pmatrix} - \theta \begin{pmatrix} x_1 \\ \theta x_1 \end{pmatrix} = \begin{pmatrix} 0 \\ -(S_2 - \theta I)r_1 + r_2 \end{pmatrix}.$$

If we use x_2/θ as a Ritz vector for (1.1), the Ritz vector for the linearization is then

$$\hat{x} = \begin{pmatrix} \theta^{-1} x_2 \\ x_2 \end{pmatrix}. \tag{4.11}$$

The residual is

$$\begin{bmatrix} 0 & I \\ S_1 & S_2 \end{bmatrix} \begin{pmatrix} \theta^{-1} x_2 \\ x_2 \end{pmatrix} - \theta \begin{pmatrix} \theta^{-1} x_2 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \theta^{-1} S_1 r_1 + r_2 \end{pmatrix}.$$

The conclusion from this analysis is that for large θ there may be an advantage in using x_2/θ and for small θ in using x_1. From (4.10) and (4.11), we derive that if x is

close enough to an eigenvector, $\|\tilde{x}\| \approx \|\hat{x}\|$. Alternatively, the norms $\|S_1\|$ and $\|S_2\|$ may also play a role in the decision to make the choice.

Related to the choice of the first or second component is the difference $x_2 - \theta x_1$. If this difference is small, it probably does not make much difference which component we take.

THEOREM 4.6. *Recall* (3.24). *Let* $x_1 = V_k z$ *and* $x_2 = W_k z$ *with* $H_k z = \theta z$ *and* $\|z\| = 1$ *and define* $\rho = \beta_k |e_k^* z|$.

$$\|x_2 - \theta x_2\|_1 \leq \xi_{k,\min}^{-1} \rho.$$

Proof. From (4.9), we have that

$$\|x_2 - \theta x_1\|_2 = \beta_k \|v_{k+1}\|_2 e_k^* z. \qquad \square$$

We conclude that $\|x_2 - \theta x_1\|_2$ is at most ρ but can be smaller. Recall that

$$\xi_{k,\min} = \sqrt{1 + \sigma_{\min}^2(\underline{H}_k)} > 1,$$

which is large when the singular values of $\underline{H}_k$ are large.

4.2.2. Schur decomposition. The eigenvectors usually do not form an orthogonal set of vectors and are not even guaranteed to exist (in the defective case). The Schur vectors always form an orthogonal basis and a Schur decomposition always exists.

For the linearized quadratic eigenvalue problem, the Schur decomposition of S is

$$S \begin{pmatrix} U \\ UT \end{pmatrix} = \begin{pmatrix} U \\ UT \end{pmatrix} T.$$

The diagonal elements of T are the Ritz values. (In the case of real matrices, T is in pseudo–upper-triangular form when a Ritz value is complex. For the details, see [6, section 7.4].)

Let $H_k Z_k = Z_k T_k$ be the Schur decomposition of H_k. Define the residual $r_k^* = \beta_k e_k^* Z_k$. The Schur vectors computed from a Q-Arnoldi triple have the form

$$\mathbf{U}_k = \mathbf{V}_k Z_k = \begin{pmatrix} V_k Z_k \\ V_k Z_k T_k + v_{k+1} r_k^* \end{pmatrix}. \tag{4.12}$$

The structure in the Schur vectors is lost in the Krylov subspace for the same reasons as in the case of the Ritz vectors. However, we can similarly select the upper or lower components as Schur vectors. When we do this, we not only add an error in the recurrence relation but also perturb the orthogonality of the basis vectors. We study this problem in detail in section 4.4.

4.3. Implicit restarting. When k gets large, the storage and computational costs of the Arnoldi method can become unacceptably high. Therefore, some form of restarting or reduction of the basis is desirable. The idea is to reduce the Krylov subspace by throwing away a part of the spectrum we are not interested in.

Implicit restarting in Arnoldi's method was first introduced by Sorensen [22]. Sorensen uses implicit QR steps. Variations on this theme have been proposed where the Schur form of the Hessenberg matrix is truncated [26, 23]. This is mathematically equivalent to [22] using exact shifts.

4.3.1. Implicit QR step. One way to perform such a reduction is called implicit restarting and was proposed by Sorensen [22]. Also see [22, 19, 11, 12, 17]. The idea is to apply an orthogonal transformation to H_k that pushes the p desired Ritz values of H_k to the principle $p \times p$ block. The orthogonal transformation produces a reduction of the subspace of dimension $k+1$ to $p+1$ keeping the p desired Ritz values.

When Z results from the QR factorization $\underline{H}_k - \mu \underline{I}$, we have that

$$\underline{Z}_k^* \underline{H}_k = R_k + \mu \underline{Z}_k^* \underline{I},$$

and so

$$\underline{Z}_k \underline{Z}_k^* \underline{H}_k \underline{Z}_{k-1} = \underline{Z}_k \left(R_k \underline{Z}_{k-1} + \mu I \right) = \underline{H}_k \underline{Z}_{k-1}.$$

So, if $\mathcal{Q} = \{V_{k+1}, \underline{H_k}, w_{k+1}\} \in \mathbf{Q}_k(S)$, following Theorem 4.5, $\mathcal{T}_Z \mathcal{Q} \in \mathbf{Q}_{p=k-1}(S)$.

4.3.2. Purging. Another way to reduce the subspace dimension is purging [11]. The idea here is to purge the undesired part of the Schur factorization of H_k.

Recall the definitions of T_k, Z_k, and r_k from section 4.2.2. By multiplying on the right by Z_k, the Arnoldi recurrence relation (3.1) can be written in terms of Schur vectors as follows:

$$S\mathbf{V}_k Z_k - \mathbf{V}_k Z_k T_k = \mathbf{v}_{k+1} r_k^*. \tag{4.13}$$

Let the Schur form be ordered so that the last $k-p$ diagonal elements in T_k are unwanted Ritz values. The idea of purging is to keep the first p Schur vectors in the basis. Removing the last $k-p$ columns from (4.13) produces

$$S\mathbf{V}_k Z_p - [\mathbf{V}_k Z_p \;\; \mathbf{v}_{k+1}] \begin{pmatrix} T_p \\ r_p^* \end{pmatrix} = 0, \tag{4.14}$$

where Z_p are the first p columns of Z, T_p is the leading $p \times p$ block of T_k, and r_p are the first p elements of r_k. There exists a unitary U_p so that

$$\underline{\tilde{H}}_p = \begin{pmatrix} U_p & 0 \\ 0 & 1 \end{pmatrix}^* \begin{pmatrix} T_p \\ r_p^* \end{pmatrix} U_p$$

is $p+1 \times p$ upper Hessenberg [6].

Let $\mathcal{T}$ have transformation matrix $\left(\begin{smallmatrix} Z_p U_p & 0 \\ 0 & 1 \end{smallmatrix}\right)$ (Definition 4.4). From

$$\underline{H}_k Z_p = \begin{bmatrix} Z_p & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} T_p \\ r_p^* \end{pmatrix},$$

we derive that

$$\begin{bmatrix} Z_p & \\ & 1 \end{bmatrix} \begin{bmatrix} Z_p & \\ & 1 \end{bmatrix}^* \underline{H}_k Z_p = \underline{H}_k Z_p,$$

and so

$$\begin{bmatrix} Z_p U_p & \\ & 1 \end{bmatrix} \begin{bmatrix} Z_p U_p & \\ & 1 \end{bmatrix}^* \underline{H}_k Z_p U_p = \underline{H}_k Z_p U_p.$$

We conclude that $\mathcal{T}\{V_{k+1}, \underline{H}_k, w_{k+1}\} \in \mathbf{Q}_p(S)$.

4.4. Locking. Suppose that the first l elements in r_k are smaller than a convergence tolerance. The idea of locking is to set these small elements explicitly to zero assuming that the Schur vectors are exact. Decompose $r_k^* = [r_l^* \;\; r_{k-l}^*]$. Locking introduces an error in the recurrence relation:

$$S\mathbf{U}_k - [\mathbf{U}_k \;\; \mathbf{v}_{k+1}] \begin{pmatrix} T_k \\ [0 \;\; r_{k-l}^*] \end{pmatrix} = \mathbf{v}_{k+1}(r_l^* \;\; 0). \tag{4.15}$$

This can be considered a backward error on the solution. The corresponding Q-Arnoldi triple is in $\mathbf{Q}(S, \|r_l\|, 0)$. Reasons for the use of locking in eigenvalue codes are the reduction of the dimension of the Krylov subspace and the computation of multiple eigenvalues without the need of block methods.

For linear problems it is accepted that the recurrence relation of Arnoldi's method has a small error. The residual term in the right-hand side of (4.15) is usually considered a backward error on S. The Bauer–Fike theorem [21, Theorem 3.6] shows an upper bound to the perturbation of the eigenvalues that is proportional to $\|r_l\|$ and that is small when the eigenvectors associated with different eigenvalues are almost orthogonal. The Schur vectors form an orthogonal set, so the orthogonality is preserved. In the following, we aim to preserve the structure of the Schur vectors (4.12). As we will see, this modifies the error on the recurrence relation and destroys the orthogonality. The goal of this section is to restore the orthogonality and analyze the impact on the recurrence relation using the results from section 4.1.3.

When we use the upper component as Schur vectors of the quadratic eigenvalue problem, the Schur vectors obtain the form

$$\tilde{\mathbf{U}}_k = \begin{pmatrix} V_k Z_l & U_{1,k-l} \\ V_k Z_l T_l & U_{2,k-l} \end{pmatrix} = \mathbf{U}_k - \begin{pmatrix} 0 & 0 \\ v_{k+1} r_l^* & 0 \end{pmatrix}. \tag{4.16}$$

Following (4.7) with $g^* = -[r_l^* \;\; 0]$, we find

$$\underline{\tilde{H}}_k = \begin{pmatrix} T_k \\ [0 \;\; r_{k-l}^*] \end{pmatrix}.$$

This is precisely the matrix we want to have with locking. Recall that $\|\tilde{G}_k\| \simeq \|G_k\| \sim \|r_l\|$ is small. The new basis can be orthogonalized by applying an appropriate transformation.

Similarly, we can use the lower part of $\mathbf{U}_k$ as Schur vectors. Decompose

$$Z_k = [Z_l \;\; Z_{k-l}] \quad \text{and} \quad T_k = \begin{bmatrix} T_l & T_{l,k-l} \\ & T_{k-l} \end{bmatrix}.$$

Now define Schur vectors using the lower part of $\mathbf{U}_l$:

$$\hat{\mathbf{U}}_k = \begin{pmatrix} V_k Z_l + v_{k+1} r_l^* T_l^{-1} & U_{1,k-l} \\ V_k Z_l T_l + v_{k+1} r_l^* & U_{2,k-l} \end{pmatrix} = \mathbf{U}_k - \begin{pmatrix} v_{k+1} r_l^* T_l^{-1} & 0 \\ 0 & 0 \end{pmatrix}. \tag{4.17}$$

This assumes that T_l is nonsingular. With $g^* = -[r_l^* T_l^{-1} \;\; 0]$, following (4.8), the structure of the vectors is restored by using

$$\underline{\hat{H}}_k = \begin{pmatrix} T_k \\ [0 \;\; \hat{r}_{k-l}^*] \end{pmatrix},$$

with $\hat{r}_{k-l}^* = r_{k-l}^* - r_l^* T_l^{-1} T_{l,k-l}$. The most important point is that the residual term corresponding to the first l Ritz values is set equal to zero. The residual terms of the remaining Ritz values are modified. This is not the case when $\tilde{\mathbf{U}}_k$ is used as the Schur basis. The error on the recurrence relation is again proportional to $\|r_l\|$.

TABLE 5.1
Illustration of instabilities in the Q-Arnoldi method. $\gamma_\perp = \|I - \mathbf{V}_{k+1}^* \mathbf{V}_{k+1}\|_F$ is the deviation of orthogonality, and γ_R is the error on the recurrence relation.

ζ	1		10^4		10^6	
$\kappa(\underline{H}_k)$	30.0		$1 \cdot 10^8$		$1 \cdot 10^{12}$	
$\|\underline{H}_k\|_2$	1.6		$1 \cdot 10^8$		$1 \cdot 10^{12}$	
$\frac{\sqrt{\gamma_k^2+\delta_k^2+1.0}}{\sqrt{k+1}}$	1.03		$3.66\,10^6$		$1.66\,10^{10}$	
	$\gamma_\perp$	γ_R	$\gamma_\perp$	γ_R	$\gamma_\perp$	γ_R
Arnoldi	$1 \cdot 10^{-13}$	$1 \cdot 10^{-16}$	$5 \cdot 10^{-14}$	$1 \cdot 10^{-16}$	$2 \cdot 10^{-13}$	$2 \cdot 10^{-16}$
Q-Arnoldi	$3 \cdot 10^{-13}$	$1 \cdot 10^{-16}$	$4 \cdot 10^{-10}$	$6 \cdot 10^{-12}$	3.0	$4 \cdot 10^{-6}$

5. Numerical examples.

5.1. Illustration of numerical instabilities in Q-Arnoldi. From section 3.4, we can see that the Q-Arnoldi method may lose stability when γ_k is large. This is possible only when $\|\underline{H}_k\|_2$ and $\kappa(\underline{H}_k)$ are large.

We have run $k = 10$ iterations of the Arnoldi method for a problem of dimension $n = 10,000$ with $K = I$, $C = \zeta I$, and $M = \zeta \mathrm{diag}(\mu_1, \ldots, \mu_n)$, where $\mu_j = 1/j$ for three values of ζ. Table 5.1 illustrates the numerical behavior of the Arnoldi and Q-Arnoldi algorithms for this example. The Q-Arnoldi algorithm is sensitive to large ζ's. The example illustrates that scaling the matrices may help improve the numerical stability of the Q-Arnoldi algorithm: indeed, the eigenvectors are the same for all cases, independent of ζ, and the eigenvalues of (1.1) are divided by ζ, but when $\zeta = 1$, K, C, and M have norms around one.

5.2. Selection of component of Ritz vectors. Consider the quadratic eigenvalue problem (1.1) with $K = I$, $M = \mathrm{diag}(\mu_1, \ldots, \mu_n)$, and $C = 0.01M$ with $\mu_j = 1/j$ for $n = 10,000$. We have run 10 steps of the Arnoldi method with an initial vector with equal components. Let $(\theta, x = \left(\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right))$ be a Ritz vector returned by the Arnoldi method. Define

$$\begin{aligned} \rho &= \|A^{-1}(\theta A - B)x\|, \\ \rho_1 &= \|K^{-1}(\theta^2 K + \theta C + M)x_1\|, \\ \rho_2 &= \|K^{-1}(\theta^2 K + \theta C + M)x_2\|. \end{aligned}$$

Table 5.2 shows the Ritz values and the corresponding residual norms. For this example, x_2 shows to be a better Ritz vector than x_1 for the large Ritz values.

5.3. Quadratic eigenvalue problem for an acoustic box. In this section we study the problem of an acoustic box with walls covered with carpet with dimensions 0.54m × 0.54m × 0.55m. The material has a complex speed of sound $340 + i3.4$, and the density is 1.225kg/m^3. The box is discretized with 64,710 hexahedral elements.

The matrices are produced by ACTRAN [5]. The problem to be solved has the form (1.1) and $n = 13,623$. We want to compute the 50 eigenvalues with imaginary part larger than 600. We applied the shift $\sigma = 600i$; see section 3.3. We used the following algorithm.

ALGORITHM 5.1 (restarted Q-Arnoldi method).

1. *Choose* $\mathbf{v}_1$ *randomly and normalize.*
2. *Until the wanted eigenvalues have converged, do:*
   2.1. *Build a Krylov subspace of dimension* k.

TABLE 5.2
Comparison of Ritz vectors.

Ritz value	ρ	ρ_1	ρ_2
1.0099	$2 \cdot 10^{-7}$	$2 \cdot 10^{-7}$	$3 \cdot 10^{-8}$
0.509808	$1 \cdot 10^{-4}$	$7 \cdot 10^{-5}$	$2 \cdot 10^{-5}$
0.342946	$3 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$6 \cdot 10^{-4}$
0.256429	$2 \cdot 10^{-2}$	$5 \cdot 10^{-3}$	$3 \cdot 10^{-3}$
0.183618	$3 \cdot 10^{-2}$	$7 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
0.117528	$3 \cdot 10^{-2}$	$4 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
-0.00203237	$7 \cdot 10^{-5}$	$1 \cdot 10^{-5}$	$5 \cdot 10^{-4}$
0.00250052	$5 \cdot 10^{-3}$	$2 \cdot 10^{-4}$	$6 \cdot 10^{-5}$
$0.0381711 \pm i0.00132763$	$3 \cdot 10^{-2}$	$2 \cdot 10^{-3}$	$2 \cdot 10^{-3}$

TABLE 5.3
$\sqrt{\gamma_k^2 + \delta_k^2}$ for the different restarts in the solution of the quadratic eigenvalue problem of the box.

Before restart	2.14777
First restart	1.93901
Second restart	1.73840
Third restart	1.69189

2.1. *Compute Ritz values, Ritz vectors, and residual norms.*
2.2. *Order the Ritz values in increasing distance to σ.*
2.3. *Purge the last $m - p$ columns of the recurrence relation.*

We solved the problem using the Arnoldi method with $k = 100$ and $p = 50$. The first iteration costs $k = 100$ products with S. In Step 2.3, the purging operation keeps p iteration vectors with Ritz values corresponding to the Ritz values nearest σ. The goal of the next iterations is to improve these values. The next call to Step 2.1 requires only $k - p$ additional iterations to obtain a subspace of dimension k. After three restarts, 50 Ritz values were computed with residual norms smaller than 10^{-8}. The computations were carried out on a Linux PC. The final loss of orthogonality in the Q-Arnoldi algorithm was

$$\|I - \mathbf{V}_{k+1}^* \mathbf{V}_{k+1}\|_F \simeq 3.1\, 10^{-13}.$$

For the Arnoldi algorithm we also had

$$\|I - \mathbf{V}_{k+1}^* \mathbf{V}_{k+1}\|_F \simeq 3.1\, 10^{-13}.$$

Table 5.3 shows $\sqrt{\gamma_k^2 + \delta_k^2}$ for the different restarts. For all restarts, $\sqrt{\gamma_k^2 + \delta_k^2}$ is small, so we do not expect numerical difficulties. This is no surprise since the shift $\sigma = 600i$ is not close to an eigenvalue of (1.1), so $\|\underline{H}_k\|_2 \leq \|S\|_2$ is small.

6. Conclusions. The Q-Arnoldi algorithm is a memory efficient implementation of the Arnoldi method for specific choices of linearization of the quadratic eigenvalue problem.

We have proposed an algorithm that preserves the structure of the Schur vectors and that shows that implicit restarting, purging, and locking similarly preserve the structure of the Arnoldi vectors.

As for the choice of linearization, due to A^{-1}, the Arnoldi method produces the same results for any linearization in $\mathbb{L}_1$. The same conclusion holds for the SOAR

method. This also implies (as we already knew) that the Arnoldi method in its standard form is not able to preserve structure.

An important conclusion lies in the influence of $\xi_{\min}$ and $\xi_{\max}$. The ratio should not be far away from one in order to reduce the chance of cancelation in the numerical computations. In addition, the components of the Ritz vectors lie in the same direction when $\xi_{\min}$ is large. However, as we mentioned earlier, for the shift-and-invert transformation, $\xi_{\min}$ usually lies close to one. We have some freedom in choosing the pole σ to keep $\xi_{\max}$ low. The derivation of scalings of S_1 and S_2 is still an open problem. Note that S_1 and S_2 are known only in factored forms $-K^{-1}C$ and $-K^{-1}M$, respectively.

The conclusion is not that the SOAR method is useless when more than one eigenvalue needs to be computed or restarting the Arnoldi process is required. The Q-Arnoldi algorithm produces a Krylov subspace, whereas the SOAR method projects K, C, and M on V_{k+1}. This is still possible in a postprocessing step in the Q-Arnoldi algorithm in order to improve the Ritz values or impose spectral structure.

The extension to higher order polynomials,

$$(A_0 + \lambda A_1 + \cdots + \lambda^p)u = 0,$$

is straightforward and leads to a similar algorithm with similar conclusions.

Acknowledgments. The author is grateful to Mickaël Robbé, who helped with an early draft of the paper. The author also thanks the anonymous referees, who improved the readability of the paper.

REFERENCES

[1] W. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[2] Z. Bai, D. Day, and Q. Ye, *ABLE: An adaptive block Lanczos method for non-Hermitian eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1060–1082.

[3] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Software Environ. Tools 11, SIAM, Philadelphia, 2000.

[4] Z. Bai and Y. Su, *SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 640–659.

[5] Free Field Technologies, *MSC.ACTRAN* 2006, *User's Manual*, Free Field Technologies, Louvain-La-Neuve, Belgium, 2006.

[6] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[7] R. Grimes, J. Lewis, and H. Simon, *A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 228–272.

[8] N. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[9] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Stand., 45 (1950), pp. 255–282.

[10] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Res. Nat. Bur. Stand., 49 (1952), pp. 33–53.

[11] R. Lehoucq and D. Sorensen, *Deflation techniques within an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.

[12] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, Software Environ. Tools 6, SIAM, Philadelphia, 1998.

[13] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Structured polynomial eigenvalue problems: Good vibrations from good linearizations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1029–1051.

[14] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 971–1004.
[15] K. Meerbergen, *Locking and restarting quadratic eigenvalue solvers*, SIAM J. Sci. Comput., 22 (2001), pp. 1814–1839.
[16] K. Meerbergen, *The rational Lanczos method for Hermitian eigenvalue problems*, Numer. Linear Algebra Appl., 8 (2001), pp. 33–52.
[17] K. Meerbergen and A. Spence, *Implicitly restarted Arnoldi and purification for the shift-invert transformation*, Math. Comp., 66 (1997), pp. 667–689.
[18] V. Mehrmann and D. Watkins, *Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils*, SIAM J. Sci. Comput., 22 (2001), pp. 1905–1925.
[19] R. Morgan, *On restarting the Arnoldi method for large nonsymmetric eigenvalue problems*, Math. Comp., 65 (1996), pp. 1213–1230.
[20] B. N. Parlett and H. C. Chen, *Use of indefinite pencils for computing damped natural modes*, Linear Algebra Appl., 140 (1990), pp. 53–88.
[21] Y. Saad, *Numerical methods for large eigenvalue problems*, Algorithms and Architectures for Advanced Scientific Computing, Manchester University Press, Manchester, UK, 1992.
[22] D. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
[23] G. W. Stewart, *A Krylov–Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 601–614.
[24] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361.
[25] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.
[26] K. Wu and H. Simon, *Thick-restart Lanczos method for large symmetric eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 602–616.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1483–1499

HOW TO MAKE SIMPLER GMRES AND GCR MORE STABLE*

PAVEL JIRÁNEK†, MIROSLAV ROZLOŽNÍK‡, AND MARTIN H. GUTKNECHT§

Abstract. In this paper we analyze the numerical behavior of several minimum residual methods which are mathematically equivalent to the GMRES method. Two main approaches are compared: one that computes the approximate solution in terms of a Krylov space basis from an upper triangular linear system for the coordinates, and one where the approximate solutions are updated with a simple recursion formula. We show that a different choice of the basis can significantly influence the numerical behavior of the resulting implementation. While Simpler GMRES and ORTHODIR are less stable due to the ill-conditioning of the basis used, the residual basis is well-conditioned as long as we have a reasonable residual norm decrease. These results lead to a new implementation, which is conditionally backward stable, and they explain the experimentally observed fact that the GCR method delivers very accurate approximate solutions when it converges fast enough without stagnation.

Key words. large-scale nonsymmetric linear systems, Krylov subspace methods, minimum residual methods, numerical stability, rounding errors

AMS subject classifications. 65F10, 65G50, 65F35

DOI. 10.1137/070707373

1. Introduction.

In this paper we consider certain methods for solving a system of linear algebraic equations

$$Ax = b, \qquad A \in \mathbb{R}^{N\times N}, \qquad b \in \mathbb{R}^N, \tag{1.1}$$

where A is a large and sparse nonsingular matrix that is, in general, nonsymmetric. For solving such systems, Krylov subspace methods are very popular. They build a sequence of iterates x_n $(n = 0, 1, 2, \ldots)$ such that $x_n \in x_0 + \mathcal{K}_n(A, r_0)$, where $\mathcal{K}_n(A, r_0) \equiv \mathrm{span}\{r_0, Ar_0, \ldots, A^{n-1}r_0\}$ is the nth Krylov subspace generated by the matrix A from the residual $r_0 \equiv b - Ax_0$ that corresponds to the initial guess x_0. Many approaches for defining such approximations x_n have been proposed; see, e.g., the books by Greenbaum [9], Meurant [16], and Saad [22]. In particular, due to their smooth convergence behavior, minimum residual methods satisfying

$$\|r_n\| = \min_{\widetilde{x}\in x_0+\mathcal{K}_n(A,r_0)} \|b - A\widetilde{x}\|, \qquad r_n \equiv b - Ax_n, \tag{1.2}$$

are widely used; see, e.g., the GMRES algorithm of Saad and Schultz [23]. We recall that the minimum residual property (1.2) is equivalent to the orthogonality condition

$$r_n \perp A\mathcal{K}_n(A, r_0),$$

*Received by the editors November 6, 2007; accepted for publication (in revised form) by D. B. Szyld August 18, 2008; published electronically December 3, 2008.
http://www.siam.org/journals/simax/30-4/70737.html

†Faculty of Mechatronics and Interdisciplinary Engineering Studies, Technical University of Liberec, Hálkova 6, CZ-461 17 Liberec, Czech Republic (pavel.jiranek@tul.cz). The work of this author was supported by the MSMT CR under the project 1M0554 "Advanced Remedial Technologies."

‡Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, CZ-182 07 Prague 8, Czech Republic (miro@cs.cas.cz). The work of this author was supported by the Grant Agency of the Czech Academy of Sciences under the projects IAA100300802 and IAA1030405 and by the Institutional Research Plan AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Applications."

§Seminar for Applied Mathematics, ETH Zurich, CH-8092 Zurich, Switzerland (mhg@math.ethz.ch).

where $\perp$ is the orthogonality relation induced by the Euclidean inner product $\langle \cdot, \cdot \rangle$.

The classical implementation of GMRES [23] makes use of a nested sequence of orthonormal bases of the Krylov subspaces $\mathcal{K}_n(A, r_0)$. These bases are generated by the Arnoldi process [2], and the approximate solution x_n satisfying the minimum residual property (1.2) is constructed from the transformed least squares problem with an upper Hessenberg matrix. This problem is solved via its recursive QR factorization, updated by applying Givens rotations. Once the norm of the residual is small enough, which can be seen without explicitly solving the least squares problem, the triangular system with the computed R-factor is solved, and the approximate solution x_n is computed. In [3, 11, 18] it was shown that this "classical" version of the GMRES method is backward stable provided that the Arnoldi process is implemented using the modified Gram–Schmidt algorithm or Householder reflections.

In this paper we deal with a different approach. Instead of building an orthonormal basis of $\mathcal{K}_n(A, r_0)$, we look for an orthonormal basis $V_n \equiv [v_1, \ldots, v_n]$ of $A\mathcal{K}_n(A, r_0)$. We will also consider a basis $Z_n \equiv [z_1, \ldots, z_n]$ of $\mathcal{K}_n(A, r_0)$ and assume in our analysis that the vectors Z_n have unit lengths, but they need not be orthogonal. The orthonormal basis V_n of $A\mathcal{K}_n(A, r_0)$ is obtained from the QR factorization of the image of Z_n:

$$AZ_n = V_n U_n. \tag{1.3}$$

Since $r_n \in r_0 + A\mathcal{K}_n(A, r_0) = r_0 + \mathcal{R}(V_n)$ and $r_n \perp \mathcal{R}(V_n)$, the residual $r_n = (I - V_n V_n^T) r_0$ is just the orthogonal projection of r_0 onto the orthogonal complement of $\mathcal{R}(V_n)$, which can be computed recursively as

$$r_n = r_{n-1} - \alpha_n v_n, \qquad \alpha_n \equiv \langle r_{n-1}, v_n \rangle \tag{1.4}$$

($\mathcal{R}(V_n)$ denotes the range of the matrix V_n). Let $R_{n+1} \equiv [r_0, \ldots, r_n]$, let $D_n \equiv \mathrm{diag}(\alpha_1, \ldots, \alpha_n)$, and let $L_{n+1,n} \in \mathbb{R}^{(n+1)\times n}$ be the bidiagonal matrix with 1's on the main diagonal and -1's on the first subdiagonal; then the recursion (1.4) can be cast into a matrix relation

$$R_{n+1} L_{n+1,n} = V_n D_n. \tag{1.5}$$

Since the columns of Z_n form a basis of $\mathcal{K}_n(A, r_0)$, we can represent x_n in the form

$$x_n = x_0 + Z_n t_n, \tag{1.6}$$

so that $r_n = r_0 - AZ_n t_n = r_0 - V_n U_n t_n$. Due to $r_n \perp \mathcal{R}(V_n)$, it follows that

$$U_n t_n = V_n^T r_0 = [\alpha_1, \ldots, \alpha_n]^T. \tag{1.7}$$

Hence, once the residual norm is small enough, we can solve this upper triangular system and compute the approximate solution $x_n = x_0 + Z_n t_n$. We call this approach the *generalized simpler approach*. Its pseudocode is given in Figure 1.1. It includes, as a special case, Simpler GMRES, which was proposed by Walker and Zhou [30], where $Z_n = [\frac{r_0}{\|r_0\|}, V_{n-1}]$. We will be also interested in the case of the residual basis $Z_n = \widetilde{R}_n \equiv [\frac{r_0}{\|r_0\|}, \ldots, \frac{r_{n-1}}{\|r_{n-1}\|}]$; we will call this case RB-SGMRES (Residual-based Simpler GMRES). Recently this method was also derived and implemented by Yvan Notay [17].

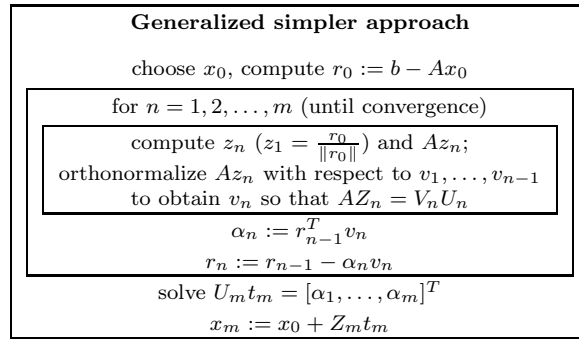


FIG. 1.1. *Pseudocode of the generalized simpler approach.*

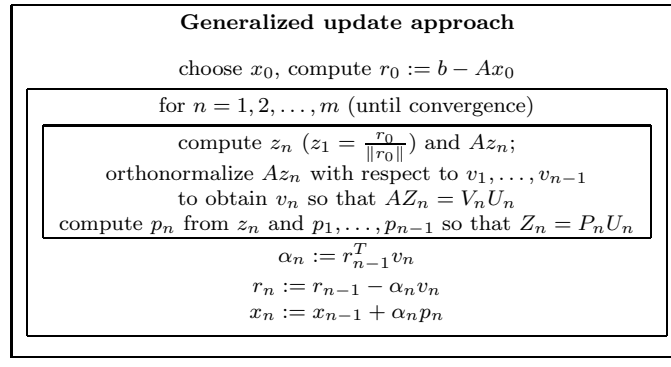


FIG. 1.2. *Pseudocode of the generalized update approach.*

Recursion (1.4) reveals the connection between the generalized simpler approach and yet another minimum residual approach. Let us set $p_k \equiv A^{-1}v_k$ $(k = 1, \ldots, n)$ and $P_n \equiv [p_1, \ldots, p_n]$. Then, left-multiplying (1.4) by A^{-1} yields

$$x_n = x_{n-1} + \alpha_n p_n \tag{1.8}$$

or, in matrix form, $X_{n+1}L_{n+1,n} = -P_nD_n$ with $X_{n+1} \equiv [x_0, \ldots, x_n]$. So, instead of computing the coordinates t_n of $x_n - x_0$ with respect to the basis Z_n, we can directly update x_n from x_{n-1}. However, this requires that we construct the direction vectors P_n forming an A^TA-orthogonal basis of $\mathcal{K}_n(A, r_0)$. Since U_n is known from (1.3), the recursion for p_n can be extracted from the formula

$$Z_n = P_nU_n. \tag{1.9}$$

Note that two recursions (1.3) and (1.9) can be run in the same loop, and we have to store all the direction vectors in P_n and all the orthonormal basis vectors in V_n. We will use the terminology *generalized update approach* for this case. Its pseudocode is given in Figure 1.2. The case $Z_n \equiv [\frac{r_0}{\|r_0\|}, V_{n-1}]$ of this method was proposed in [20] under the name A^TA-variant of GMRES, and up to the normalization of the vectors V_n in (1.3) it is equivalent to the ORTHODIR algorithm due to Young and Jea [33, 7]. Likewise, the case $Z_n = [r_0, \ldots, r_{n-1}]$ corresponds to the GCR (or full ORTHOMIN) method of Elman, Eisenstat, and Schultz [6, 5] (the orthogonal vectors v_n are unnormalized in the original implementation), and it is identical to

TABLE 1.1
Computational costs (without the cost of $m+1$ matrix-vector products) and storage requirements (without the storage of A) of the generalized simpler and update approaches after m iteration steps.

	Computational costs	Storage requirements
Generalized simpler approach	$(2N+\frac{1}{2})m^2+(9N-\frac{1}{2})m+4N$	$(2N+\frac{3}{2})m+\frac{1}{2}m^2+2N+1$
Generalized update approach	$(3N-\frac{1}{2})m^2+(9N-\frac{1}{2})m+4N$	$(2N+1)m+2N+2$

the GMRESR method [28] of van der Vorst and Vuik (with the choice $u_n^{(0)} = r_n$). Without normalization it was also treated in [33]. As we have already mentioned, here we will analyze the choice $Z_n = \widetilde{R}_n$. The importance of normalizing Z_n before the orthogonalization in (1.3) will be seen later.

In Table 1.1 we summarize the computational costs and storage requirements of performing m iteration steps in the generalized simpler approach and the generalized update approach, where we have excluded the storage for A and the cost of $m+1$ matrix-vector products. In both approaches we have to store two sets of vectors—the bases V_m and Z_m (the generalized simpler approach) or V_m and P_m (the generalized update approach)—making these schemes comparable to FGMRES [21], the (flexible) preconditioned variant of the standard GMRES method [23]. This remains true also in the case of preconditioned versions of our algorithms, but we do not treat these explicitly here. In contrast to the generalized simpler approach, we do not need to store the triangular $m \times m$ matrix of orthogonalization coefficients U_m in the generalized update approach, but we have to compute the additional set of vectors P_m. Some savings are possible in special cases, as in Simpler GMRES with the particular choice of the basis $Z_m = [\frac{r_0}{\|r_0\|}, V_{m-1}]$, where the last $m-1$ columns of Z_m need not to be stored and normalized again. Simpler GMRES is in terms of work and storage competitive to the GMRES method, which in addition was shown to be backward stable and in this context should clearly be the method of choice when preconditioning is not considered.

The paper is organized as follows. In section 2 we analyze first the maximum attainable accuracy of the generalized simpler approach based on (1.6) and (1.7). Then we turn to the generalized update approach based on (1.9) and (1.8). To keep the text readable, we assume rounding errors only in selected, most relevant parts of the computation. The bounds presented in Theorems 2.1 and 2.3 show that the conditioning of the matrix Z_n plays an important role in the numerical stability of these schemes. Both theorems give bounds on the maximum attainable accuracy measured by the normwise backward error. We also formulate analogous statements for the residual norm in terms of the condition number of the matrix U_n. While for the generalized simpler approach these bounds do not depend on the conditioning of A, the bound for the generalized update approach is proportional to $\kappa(A)$ (as we will show in our constructed numerical example, the bound is attained). However, the additional factor of $\kappa(A)$ in the generalized update approach is usually an overestimate; in practice, both approaches behave almost equally well for the same choice of basis. This is especially true for the relative errors of the computed approximate solutions, where we have essentially the same upper bound. The situation is completely analogous to results for the MINRES method [19] given by Sleijpen, van der Vorst, and Modersitzki in [25].

In section 3 we derive particular results for two choices of the basis Z_n—first for $Z_n = [\frac{r_0}{\|r_0\|}, V_{n-1}]$, leading to Simpler GMRES by Walker and Zhou [30] and to ORTHODIR, and then for $Z_n = \widetilde{R}_n$, which leads to RB-SGMRES and to a variant of GCR, respectively. It turns out that the two choices lead to a truly different behavior in the condition number of U_n, which governs the stability of the considered schemes. Since all these methods converge in a finite number of iterations, we fix the iteration index n such that $r_0 \notin A\mathcal{K}_{n-1}(A, r_0)$; that is, the exact solution has not yet been reached. Based on this we give conditions on the linear independence of the basis Z_n. It is known that the residuals are linearly dependent (or even identical) when the GMRES method stagnates (a breakdown occurs in GCR as well as in RB-SGMRES), while this does not happen for $[\frac{r_0}{\|r_0\|}, V_{n-1}]$ (Simpler GMRES and ORTHODIR are breakdown-free). On the other hand, we show that while the choice $Z_n = [\frac{r_0}{\|r_0\|}, V_{n-1}]$ leads to inherently unstable or numerically less stable schemes, the second selection $Z_n = \widetilde{R}_n$ gives rise to conditionally stable implementations provided that we have some reasonable residual decrease. In particular, we show that the RB-SGMRES implementation is conditionally backward stable. Our theoretical results are illustrated by selected numerical experiments. In section 4 we draw conclusions and give directions for future work.

Throughout the paper, we denote by $\|\cdot\|$ the Euclidean vector norm and the induced matrix norm and by $\|\cdot\|_F$ the Frobenius norm. Moreover, for $B \in \mathbb{R}^{N\times n}$ $(N \geq n)$ of rank n, $\sigma_1(B) \geq \sigma_n(B) > 0$ are the extremal singular values of B, and $\kappa(B) = \sigma_1(B)/\sigma_n(B)$ is the spectral condition number. By I we denote the unit matrix of a suitable dimension and by e_k $(k = 1, 2, \ldots)$ its kth column, and we let $e \equiv [1, \ldots, 1]^T$. We assume the standard model of finite precision arithmetic with the unit roundoff u (see Higham [13] for details). In our bounds, instead of distinguishing between several constants (which are in fact low-degree polynomials in N and n that can differ from place to place), we use the generic name c for constants.

2. Maximum attainable accuracy of the generalized simpler and update approaches. In this section we analyze the final accuracy level of the generalized simpler and update approaches formulated in the previous section. In order to make our analysis readable, we assume that only the computations performed in (1.3), (1.7), and (1.9) are affected by rounding errors.

Different orthogonalization techniques for computing the columns of V_n can be applied in the QR factorization (1.3). Here we focus on such implementations where the computed R-factor U_n has been obtained in a backward stable way; i.e., there exists an orthonormal matrix $\hat{V}_n$ so that $\hat{V}_n$ and V_n satisfy

$$AZ_n = \hat{V}_n U_n + E_n, \qquad \|E_n\| \leq cu\|A\|\|Z_n\|, \tag{2.1}$$

$$AZ_n = V_n U_n + F_n, \qquad \|F_n\| \leq cu\|A\|\|Z_n\|. \tag{2.2}$$

This is certainly true for the implementation based on Householder reflections [32], the modified Gram–Schmidt process [18], or the Gram–Schmidt process with full reorthogonalization [3]. For details we refer the reader to [13, 8]. From [31, 13] we have for the computed solution $\hat{t}_n$ of (1.7) that

$$(U_n + \Delta U_n)\hat{t}_n = D_n e, \qquad |\Delta U_n| \leq cu|U_n|, \tag{2.3}$$

where the absolute value and inequalities are understood componentwise. The approximation $\hat{x}_n$ to x is then computed as

$$\hat{x}_n = x_0 + Z_n\hat{t}_n. \tag{2.4}$$

The crucial quantity for the analysis of the maximum attainable accuracy is the gap between the true residual $b - A\hat{x}_n$ of the computed approximation and the updated residual r_n obtained from the update formula (1.4) describing the projection of the previous residual; see [9, 12]. In fact, once the updated residual becomes negligible compared to the true one (and in all algorithms considered here it ultimately will), the gap will be equal to the true residual divided by $\|A\|\|\hat{x}_n\|$, which therefore can be thought of as the normwise backward error of the ultimate approximate solution $\hat{x}_n$ (after suitable normalization). Here is our basic result on this gap for the generalized simpler approach.

THEOREM 2.1. *In the generalized simpler approach, if* $cu\kappa(A)\kappa(Z_n) < 1$*, the gap between the true residual* $b - A\hat{x}_n$ *and the updated residual* r_n *satisfies*

$$\frac{\|b - A\hat{x}_n - r_n\|}{\|A\|\|\hat{x}_n\|} \leq cu\kappa(Z_n)\left(1 + \frac{\|x_0\|}{\|\hat{x}_n\|}\right).$$

Proof. From (2.4), (2.2), and (2.3) we have $b - A\hat{x}_n = r_0 - AZ_n\hat{t_n} = r_0 - (V_nU_n + F_n)(U_n + \Delta U_n)^{-1}D_ne$, and (1.4) gives $r_n = r_0 - V_nD_ne$. It is clear from (2.1) and (2.3) that the assumption $cu\kappa(A)\kappa(Z_n) < 1$ implies the invertibility of the perturbed matrix $U_n + \Delta U_n$. Using the identity $I - U_n(U_n + \Delta U_n)^{-1} = \Delta U_n(U_n + \Delta U_n)^{-1}$ and the relation $Z_n(U_n + \Delta U_n)^{-1}D_ne = Z_n\hat{t}_n = \hat{x}_n - x_0$ following from (2.4) and (2.3), we can express the gap between $b - A\hat{x}_n$ and r_n as

$$\begin{aligned}
b - A\hat{x}_n - r_n &= (V_n - (V_nU_n + F_n)(U_n + \Delta U_n)^{-1})D_ne \\
&= (V_n(I - U_n(U_n + \Delta U_n)^{-1}) - F_n(U_n + \Delta U_n)^{-1})D_ne \\
&= (V_n\Delta U_n - F_n)(U_n + \Delta U_n)^{-1}D_ne \\
&= (V_n\Delta U_n - F_n)Z_n^{\dagger}Z_n(U_n + \Delta U_n)^{-1}D_ne \\
&= (V_n\Delta U_n - F_n)Z_n^{\dagger}(\hat{x}_n - x_0).
\end{aligned}$$

Taking the norm, considering (2.1), and noting that the terms in $V_n\Delta U_n$ and F_n can be subsumed into the generic constant c, we get $\|V_n\Delta U_n - F_n\| \leq cu\|A\|\|Z_n\|$ and

$$\|b - A\hat{x}_n - r_n\| \leq cu\|A\|\kappa(Z_n)\|\hat{x}_n - x_0\|.$$

Using the triangle inequality and division by $\|A\|\|\hat{x}_n\|$ concludes the proof. ∎

In the previous theorem we have expressed the residual gap using the difference between the actual and initial approximations $\hat{x}_n$ and x_0, respectively. However, its norm is strongly influenced by the conditioning of the upper triangular matrix U_n. As shown in section 3, the matrix U_n can be ill-conditioned for the particular case $Z_n = [\frac{r_0}{\|r_0\|}, V_{n-1}]$, thus leading to an inherently unstable scheme, whereas (under some assumptions) the scheme with $Z_n = \widetilde{R}_n$ gives rise to a well-conditioned triangular matrix U_n. In the following corollary we give a bound for the residual gap in terms of the minimal singular values of the matrices Z_k and norms of the updated residuals r_{k-1}, $k = 1, \ldots, n$.

COROLLARY 2.2. *In the generalized simpler approach, if* $cu\kappa(A)\kappa(Z_n) < 1$*, the gap between the true residual* $b - A\hat{x}_n$ *and the updated residual* r_n *satisfies*

$$\|b - A\hat{x}_n - r_n\| \leq \frac{cu\kappa(A)}{1 - cu\kappa(A)\kappa(Z_n)} \sum_{k=1}^{n} \frac{\|r_{k-1}\|}{\sigma_k(Z_k)}.$$

Proof. The gap between the true residual $b-A\hat{x}_n$ and the updated residual r_n can be expressed as $b-A\hat{x}_n-r_n=(V_n\Delta U_n-F_n)(U_n+\Delta U_n)^{-1}D_ne$. Since $e_k^TD_ne_k=\alpha_k$ and $|\alpha_k|=\sqrt{\|r_{k-1}\|^2-\|r_k\|^2}\le\sqrt{2}\|r_{k-1}\|$, the norm of the term $(U_n+\Delta U_n)^{-1}D_ne$ can be estimated as follows:

$$\begin{aligned}\|(U_n+\Delta U_n)^{-1}D_ne\| &\le \sum_{k=1}^{n}\|(U_n+\Delta U_n)^{-1}D_ne_k\| \\ &\le \sqrt{2}\sum_{k=1}^{n}\frac{\|r_{k-1}\|}{\sigma_k([U_n+\Delta U_n]_{1:k,1:k})},\end{aligned} \tag{2.5}$$

where $[U_n+\Delta U_n]_{1:k,1:k}$ denotes the principal $k\times k$ submatrix of $U_n+\Delta U_n$. Owing to (2.4), we can estimate the perturbation of $[U_n]_{1:k,1:k}=U_k$ as $\|[\Delta U_n]_{1:k,1:k}\|\le cu\|U_k\|$. Perturbation theory of singular values (see, e.g., [14]) shows that

$$\sigma_k([U_n+\Delta U_n]_{1:k,1:k})\ge\sigma_k(U_k)-cu\|U_k\|\ge\sigma_k(AZ_k)-cu\|A\|\|Z_k\| \tag{2.6a}$$

$$\ge\sigma_N(A)\sigma_k(Z_k)-cu\|A\|\|Z_k\|, \tag{2.6b}$$

which together with (2.5) concludes the proof. ∎

The estimates (2.5) and (2.6a) given in the previous proof that involve the minimum singular values of U_k $(k=1,\dots,n)$ are quite sharp. However, the estimate (2.6b) relating the minimum singular values of U_k to those of Z_k can be a large underestimate, as also observed in our numerical experiments in section 3.

Next we analyze the maximum attainable accuracy of the generalized update approach. We assume that in finite precision arithmetic the computed direction vectors satisfy

$$Z_n=P_nU_n+G_n,\qquad \|G_n\|\le cu\|P_n\|\|U_n\|. \tag{2.7}$$

This follows from the standard rounding error analysis of the recursion for vectors P_n. Note that the norm of the matrix G_n cannot be bounded by $cu\|A\|\|Z_n\|$ as it can in the case of the QR factorization (2.2). We update then the approximate solution $\hat{x}_n$ according to (1.8):

$$\hat{x}_n=\hat{x}_{n-1}+\alpha_np_n. \tag{2.8}$$

THEOREM 2.3. *In the generalized update approach, if $cu\kappa(A)\kappa(Z_n)<1$, the gap between the true residual $b-A\hat{x}_n$ and the updated residual r_n satisfies*

$$\frac{\|b-A\hat{x}_n-r_n\|}{\|A\|\|\hat{x}_n\|}\le\frac{cu\kappa(A)\kappa(Z_n)}{1-cu\kappa(A)\kappa(Z_n)}\left(1+\frac{\|x_0\|}{\|\hat{x}_n\|}\right).$$

Proof. From (2.8), (1.4), (2.2), and (2.7), $\hat{x}_n=x_0+P_nD_ne=x_0+(Z_n-G_n)U_n^{-1}D_ne$ and $r_n=r_0-V_nD_ne=r_0-(AZ_n-F_n)U_n^{-1}D_ne$, we have that

$$b-A\hat{x}_n-r_n=(AG_n-F_n)U_n^{-1}D_ne, \tag{2.9}$$

and from (2.7) and (2.1) we get $P_n=A^{-1}\hat{V}_n+A^{-1}E_nU_n^{-1}-G_nU_n^{-1}$. The norm of the matrix G_n in (2.7) can hence be bounded by

$$\|G_n\|\le cu\kappa(A)\|Z_n\|. \tag{2.10}$$

Owing to (2.8), we have the identity $U_n^{-1}D_n e = U_n^{-1}P_n^{\dagger}P_n D_n e = (P_nU_n)^{\dagger}(\hat{x}_n - x_0)$, where $\|(P_nU_n)^{\dagger}\| \le [1 - cu\kappa(A)\kappa(Z_n)]^{-1}\|Z_n^{\dagger}\|$, as follows from (2.7). Thus we obtain

$$\|U_n^{-1}D_n e\| \le \frac{\|Z_n^{\dagger}\|}{1 - cu\kappa(A)\kappa(Z_n)}\|\hat{x}_n - x_0\|, \tag{2.11}$$

which together with (2.9), (2.10), and (2.2) leads to

$$\|b - A\hat{x}_n - r_n\| \le \frac{cu\|A\|\kappa(A)\kappa(Z_n)}{1 - cu\kappa(A)\kappa(Z_n)}\|\hat{x}_n - x_0\|.$$

The proof is concluded using the triangle inequality and dividing by $\|A\|\|\hat{x}_n\|$. ∎

In the following we formulate an analogous corollary for the residual gap as in the case of the generalized simpler approach.

COROLLARY 2.4. *In the generalized update approach, if* $cu\kappa(A)\kappa(Z_n) < 1$, *the gap between the true residual* $b - A\hat{x}_n$ *and the updated residual* r_n *satisfies*

$$\|b - A\hat{x}_n - r_n\| \le \frac{cu\kappa^2(A)}{1 - cu\kappa(A)\kappa(Z_n)}\sum_{k=1}^{n}\frac{\|r_{k-1}\|}{\sigma_k(Z_k)}.$$

Proof. Considering (2.2), (2.7), and (2.10) the norm of the term $AG_n - F_n$ in (2.9) can be bounded as $\|AG_n - F_n\| \le cu\|A\|\kappa(A)$, while the term $U_n^{-1}D_n e$ can be treated as in Corollary 2.2. ∎

The bound on the ultimate backward error given in Theorem 2.3 is worse than the one in Theorem 2.1. We see that for the generalized simpler approach the normwise backward error is of the order of the roundoff unit, whereas for the generalized update approach we have an upper bound proportional to the condition number of A. Similarly, the bounds on the ultimate relative residual norms given in Corollaries 2.2 and 2.4 indicate that the relative residuals in the generalized simpler approach will reach the level which is approximately equal to $u\kappa(A)$, while in the generalized update approach this level becomes $u\kappa^2(A)$.

In the previous text we have given bounds in terms of the true residual $b - A\hat{x}_n$ and the updated residual r_n. It should be noted that the true residual is not available in practical computations, but for verification or for other purposes it can be estimated by the explicit evaluation of $\mathrm{fl}(b - A\hat{x}_n)$. It is clear from $\|\mathrm{fl}(b - A\hat{x}_n) - (b - A\hat{x}_n)\| \le cu(\|b\| + \|A\|\|\hat{x}_n\|) \le cu\|A\|(\|x\| + \|\hat{x}_n\|)$ that the error in the evaluation of the true residual (if needed) is significantly smaller than other quantities involved in our analysis.

In Theorems 2.1 and 2.3 we have estimated the attainable level of the normwise backward error of both generalized simpler and update approaches. The resulting bound is in general worse for the generalized update approach. However, as shown below, it appears that *the generalized update approach leads to an approximate solution whose forward error is essentially on the same accuracy level as the generalized simpler approach.* A similar phenomenon was also observed by Sleijpen, van der Vorst, and Modersitzki [25] in the symmetric case for two different implementations (called GMRES and MINRES in their paper).

COROLLARY 2.5. *If* $cu\kappa(A)\kappa(Z_n) < 1$, *the gap between the error* $x - \hat{x}_n$ *and the vector* $A^{-1}r_n$ *in both the generalized simpler and update approaches satisfies*

$$\frac{\|(x - \hat{x}_n) - A^{-1}r_n\|}{\|x\|} \le \frac{cu\kappa(A)\kappa(Z_n)}{1 - cu\kappa(A)\kappa(Z_n)}\frac{\|\hat{x}_n\| + \|x_0\|}{\|x\|}.$$

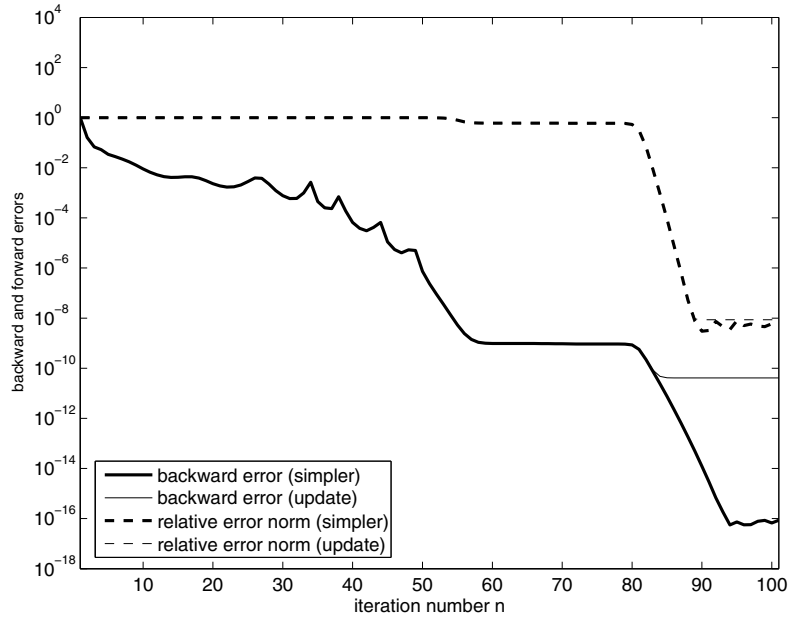


FIG. 2.1. *The test problem solved by the generalized simpler and update approaches with the almost orthonormal basis Z_n satisfying $\kappa(Z_n) \approx 1$.*

Proof. For the generalized simpler approach, the result follows directly from Theorem 2.1. For the generalized update approach, using (2.9) we have

$$(x - \hat{x}_n) - A^{-1}r_n = (-A^{-1}F_n + G_n)U_n^{-1}D_n e,$$

and the statement follows from (2.2), (2.10), and (2.11). ∎

Theorems 2.1 and 2.3 indicate that *as soon as the backward error of the approximate solution in the generalized simpler approach gets below $cu\kappa(A)\kappa(Z_n)$, the difference between the backward errors in the generalized simpler and update approaches may become visible and can be expected to be up to the order of $\kappa(A)$.* Based on our experience it is difficult to find an example where this difference is significant. Similarly to Sleijpen, van der Vorst, and Modersitzki [25], we use here a model example, where $A = G_1 D G_2^T \in \mathbb{R}^{100\times 100}$ with $D = \mathrm{diag}(10^{-8}, 2\cdot 10^{-8}, 3, 4, \ldots, 100)$ and with G_1 and G_2 being Givens rotations over the angle of $\frac{\pi}{4}$ in the $(1, 10)$-plane and the $(1, 100)$-plane, respectively; finally, $b = e$. The numerical experiments were performed in MATLAB using double precision arithmetic ($u \approx 10^{-16}$) and $x_0 = 0$. In Figure 2.1 we have plotted the normwise backward errors $\|b - A\hat{x}_n\|/(\|A\|\|\hat{x}_n\|)$ (thin and thick solid lines), and the relative 2-norms of the errors $\|x - \hat{x}_n\|/\|x\|$ (thin and thick dash-dotted lines). In all our experiments the basis V_n in (1.3) is computed with the modified Gram–Schmidt orthogonalization process, where the upper triangular factor U_n is obtained in a backward stable way satisfying (2.1). In order to ensure that the difference is not affected by a possibly high condition number of Z_n, we use the implementation where the basis Z_n is computed with the modified Gram–Schmidt Arnoldi process so that $\kappa(Z_n) \approx 1$. We see that the actual backward errors are close to each other until they stagnate: for the generalized update approach this happens approximately at a level approaching $u\kappa(A)$, while for the generalized simpler

approach we have stagnation on the roundoff unit level u. Similar observations could be made for the relative true residual norms (for better readability they are not shown in Figure 2.1); in the case of the generalized simpler approach the final level of the relative 2-norm of the true residual is on the level of $u\kappa(A)$, while for the generalized update approach this level is approximately one factor of $\kappa(A)$ higher. In contrast, the 2-norms of the errors stagnate on the $u\kappa(A)$ level in both approaches considered.

3. Choice of basis and stability. In this section we discuss the two main particular choices for the matrix Z_n leading to different algorithms for the generalized simpler and update approaches. For the sake of simplicity, we assume exact arithmetic here. The conditioning of Z_n plays an important role in our analysis. The effect of scaling the columns on the condition number has been analyzed by van der Sluis in [27], who showed that the normalization of columns is a nearly optimal strategy producing the condition number within the factor $\sqrt{n}$ of the minimum 2-norm condition number achievable by column scaling.

First, we choose $Z_n = [\frac{r_0}{\|r_0\|}, V_{n-1}]$, which leads to the Simpler GMRES method of Walker and Zhou [30] and to ORTHODIR by Young and Jea [33]. Hence, we choose $\{\frac{r_0}{\|r_0\|}, v_1, \ldots, v_{n-1}\}$ as a basis of $\mathcal{K}_n(A, r_0)$. To be sure that such a choice is adequate, we state the following simple lemma.

LEMMA 3.1. *Let $v_1, \ldots, v_{n-1}$ be an orthonormal basis of $A\mathcal{K}_{n-1}(A, r_0)$, $r_0 \notin A\mathcal{K}_{n-1}(A, r_0)$. Then the vectors $\frac{r_0}{\|r_0\|}, v_1, \ldots, v_{n-1}$ form a basis of $\mathcal{K}_n(A, r_0)$.*

Proof. The result follows easily from the assumption $r_0 \notin A\mathcal{K}_{n-1}(A, r_0)$. ∎

Note that if $r_0 \in A\mathcal{K}_n(A, r_0)$, then the condition (1.2) yields $x_n = A^{-1}b$, $r_n = 0$, and any implementation of a minimum residual method will terminate. Lemma 3.1 ensures that it makes sense to build an orthonormal basis V_n of $A\mathcal{K}_n(A, r_0)$ by the successive orthogonalization of the columns of the matrix $A[\frac{r_0}{\|r_0\|}, V_{n-1}]$ via (1.3). It reflects the fact that, for any initial residual r_0, both Simpler GMRES and ORTHODIR converge (in exact arithmetic) to the exact solution; see [33]. However, as observed by Liesen, Rozložník, and Strakoš [15], this choice of the basis is not very suitable from the stability point of view. This shortcoming is reflected by the unbounded growth of the condition number of $[\frac{r_0}{\|r_0\|}, V_{n-1}]$ discussed next. The upper bound we recall here was also derived in [30].

THEOREM 3.2. *Let $r_0 \notin A\mathcal{K}_{n-1}(A, r_0)$. Then the condition number of $[\frac{r_0}{\|r_0\|}, V_{n-1}]$ satisfies*

$$\frac{\|r_0\|}{\|r_{n-1}\|} \le \kappa([\tfrac{r_0}{\|r_0\|}, V_{n-1}]) \le 2\frac{\|r_0\|}{\|r_{n-1}\|}.$$

Proof. Since $r_{n-1} = (I - V_{n-1}V_{n-1}^T)r_0$, it is easy to see that r_{n-1} is the residual of the least squares problem $V_{n-1}y \approx r_0$. The statement therefore follows from [15, Theorem 3.2]. ∎

The conditioning of $[\frac{r_0}{\|r_0\|}, V_{n-1}]$ is thus related to the convergence of the method; in particular, it is inversely proportional to the actual relative norm of the residual. Small residuals lead to the ill-conditioning of the matrices $A[\frac{r_0}{\|r_0\|}, V_{n-1}]$ and U_n, and this negatively affects the accuracy of computed approximate solutions. This essentially means that, after some initial residual reduction, Simpler GMRES and ORTHODIR can behave unstably, which makes our analysis on maximum attainable accuracy inapplicable.

As a remedy, we now turn to the second choice, $Z_n = \widetilde{R}_n$, which leads to RB-SGMRES (proposed here as a more stable counterpart of Simpler GMRES) and to a

version of GCR due to Eisenstat, Elman, and Schultz [6, 5] (see also [29]). Hence, we choose normalized residuals $r_0, \ldots, r_{n-1}$ as the basis of $\mathcal{K}_n(A, r_0)$. To make sure that such a choice is adequate, we state the following result.

LEMMA 3.3. *Let $v_1, \ldots, v_{n-1}$ be an orthonormal basis of $A\mathcal{K}_{n-1}(A, r_0)$, $r_0 \notin A\mathcal{K}_{n-1}(A, r_0)$, and $r_k = (I - V_k V_k^T) r_0$, where $V_k \equiv [v_1, \ldots, v_k]$, $k = 1, 2, \ldots, n-1$. Then the following statements are equivalent:*

1. *$\|r_k\| < \|r_{k-1}\|$ for all $k = 1, \ldots, n-1$,*
2. *$r_0, \ldots, r_{n-1}$ are linearly independent.*

Proof. Since $r_0 \notin A\mathcal{K}_{n-1}(A, r_0) = \mathcal{R}(V_{n-1})$, we have $r_k \neq 0$ for all $k = 0, 1, \ldots, n-1$. It is clear that $\|r_k\| < \|r_{k-1}\|$ if and only if $\langle r_{k-1}, v_k \rangle \neq 0$. If that holds for all $k = 1, \ldots, n-1$, the diagonal matrix D_{n-1} is nonsingular. Using the relation (1.5), we find that $R_n[L_{n,n-1}, e_n] = [V_{n-1} D_{n-1}, r_{n-1}]$. Since $r_{n-1} \perp V_{n-1}$, the matrix $[V_{n-1} D_{n-1}, r_{n-1}]$ has orthogonal nonzero columns, and hence its rank equals n. Moreover, $\operatorname{rank}([L_{n,n-1}, e_n]) = n$, and thus $\operatorname{rank}(R_n) = n$; i.e., $r_0, \ldots, r_{n-1}$ are linearly independent. Conversely, from the same matrix relation we find that if $r_0, \ldots, r_{n-1}$ are linearly independent, then $\operatorname{rank}([V_{n-1} D_{n-1}, r_{n-1}]) = n$, and hence D_{n-1} is nonsingular, which proves that $\|r_k\| < \|r_{k-1}\|$ for all $k = 1, \ldots, n-1$. ∎

Therefore, if the method does not stagnate, i.e., if the 2-norms of the residuals $r_0, \ldots, r_{n-1}$ are strictly monotonously decreasing, then $r_0, \ldots, r_{n-1}$ are linearly independent. In this case, we can build an orthonormal basis V_n of $A\mathcal{K}_n(A, r_0)$ by the successive orthogonalization of the columns of $A\widetilde{R}_n$ via (1.3). If $r_0 \in A\mathcal{K}_{n-1}(A, r_0)$, we have an exact solution of (1.1), and the method terminates with $x_{n-1} = A^{-1} b$.

Several conditions for the nonstagnation of the minimum residual method have been given in the literature. For example, Eisenstat, Elman, and Schultz [5, 6] show that GCR (and hence any minimum residual method) does not stagnate if the symmetric part of A is positive definite, i.e., if the origin is not contained in the field of values of A. See also Greenbaum and Strakoš [10] for a different proof and Eiermann and Ernst [4]. Several other conditions can be found in Simoncini and Szyld [24] and the references therein. If stagnation occurs, the residuals are no longer linearly independent, and thus the method prematurely breaks down. In particular, if $0 \in \mathcal{F}(A)$, choosing x_0 such that $\langle A r_0, r_0 \rangle = 0$ leads to a breakdown in the first step. This was first pointed out by Young and Jea [33] with a simple 2×2 example.

However, as shown in the following theorem, when the minimum residual method does not stagnate, the columns of $\widetilde{R}_n$ are a reasonable choice for the basis of $\mathcal{K}_n(A, r_0)$.

THEOREM 3.4. *If $r_0 \notin A\mathcal{K}_{n-1}(A, r_0)$ and $\|r_k\| < \|r_{k-1}\|$ for all $k = 1, \ldots, n-1$, the condition number of $\widetilde{R}_n$ satisfies*

$$1 \leq \kappa(\widetilde{R}_n) \leq \sqrt{n}\, \gamma_n, \qquad \gamma_n \equiv \sqrt{1 + \sum_{k=1}^{n-1} \frac{\|r_{k-1}\|^2 + \|r_k\|^2}{\|r_{k-1}\|^2 - \|r_k\|^2}}. \tag{3.1}$$

Proof. From (1.5) it follows that

$$\widetilde{R}_n[\widetilde{L}_{n,n-1}, e_n] = \left[V_{n-1}, \frac{r_{n-1}}{\|r_{n-1}\|} \right], \quad \widetilde{L}_{n,n-1} \equiv \operatorname{diag}(\|r_0\|, \ldots, \|r_{n-1}\|) L_{n,n-1} D_{n-1}^{-1}.$$

Since $[V_{n-1}, \frac{r_{n-1}}{\|r_{n-1}\|}]$ is an orthonormal matrix, we have from [14, Theorem 3.3.16]

$$\begin{aligned} 1 = \sigma_n \left(\left[V_{n-1}, \frac{r_{n-1}}{\|r_{n-1}\|} \right] \right) &\leq \sigma_n(\widetilde{R}_n) \|[\widetilde{L}_{n,n-1}, e_n]\| \\ &\leq \sigma_n(\widetilde{R}_n) \|[\widetilde{L}_{n,n-1}, e_n]\|_F. \end{aligned}$$

The value of $\|[\widetilde{L}_{n,n-1}, e_n]\|_F$ can be directly computed as

$$\|[\widetilde{L}_{n,n-1}, e_n]\|_F = \sqrt{1 + \sum_{k=1}^{n-1} \frac{\|r_{k-1}\|^2 + \|r_k\|^2}{\|r_{k-1}\|^2 - \|r_k\|^2}} = \gamma_n$$

since $\alpha_k^2 = \|r_{k-1}\|^2 - \|r_k\|^2$. The statement then follows using $\|\widetilde{R}_n\| \leq \|\widetilde{R}_n\|_F \leq \sqrt{n}$. □

We define the quantity γ_n in (3.1) as the *stagnation factor*. The conditioning of $\widetilde{R}_n$ is thus related to the convergence of the method, but in contrast to the conditioning of $[\frac{r_0}{\|r_0\|}, V_{n-1}]$, it is related to the intermediate decrease of the residual norms and not to the residual decrease with respect to the initial residual. A different bound for the conditioning of the matrix $\widetilde{R}_n$ in terms of the residual norms of GMRES and FOM could be derived using the approach in [26].

We illustrate our theoretical results by two numerical examples using the ill-conditioned matrices FS1836 ($\|A\| \approx 1.2 \cdot 10^9$, $\kappa(A) \approx 1.5 \cdot 10^{11}$) and STEAM1 ($\|A\| \approx 2.2 \cdot 10^7$, $\kappa(A) \approx 3 \cdot 10^7$) obtained from the Matrix Market [1] with the right-hand side $b = Ae$ and with the initial guess $x_0 = 0$. In Figures 3.1, 3.2, 3.4, and 3.5 we show the normwise backward error $\|b - Ax_n\|/(\|A\|\|x_n\|)$, the relative norm of the residual $\|b - Ax_n\|/\|b\|$ and $\|r_n\|/\|b\|$, and the relative norms of the error $\|x - x_n\|/\|x\|$ for the choice $Z_n = [\frac{r_0}{\|r_0\|}, V_{n-1}]$ that corresponds to Simpler GMRES and ORTHODIR (Figures 3.1 and 3.4), and for $Z_n = \widetilde{R}_n$ corresponding to RB-SGMRES and GCR (Figures 3.2 and 3.5), respectively. In Figures 3.3 and 3.6 we report the condition numbers of the system matrix A, the basis Z_n, and the triangular matrix U_n multiplied by the unit roundoff u. We see that the backward errors, residual norms, and error norms are almost identical for corresponding implementations of the generalized simpler and update approaches. This can be observed in most cases: the differences between Simpler GMRES and ORTHODIR, and RB-SGMRES and GCR, respectively, are practically negligible. Figures 3.1 and 3.4 illustrate our theoretical considerations and show that, after some initial reduction, *the backward error of Simpler GMRES and ORTHODIR may stagnate at a significantly higher level than the backward error of RB-SGMRES or GCR, which stagnates at a level proportional to the roundoff units*, as shown in Figures 3.2 and 3.5. Due to Theorem 3.2, after some initial phase, the norms of the errors start to diverge in Simpler GMRES and ORTHODIR, while for RB-SGMRES and GCR we have a stagnation on a level approximately proportional to $u\kappa(A)$. The difference is clearly caused by the choice of the basis Z_n, which has an effect on the conditioning of the matrix U_n. We see that $\widetilde{R}_n$ remains well-conditioned up to the very end of the iteration process, while the conditioning of $[\frac{r_0}{\|r_0\|}, V_{n-1}]$ is linked to the convergence of Simpler GMRES and may lead to a very ill-conditioned triangular matrix U_n. Consequently, the approximate solution x_n computed from (1.7) becomes inaccurate, and its error starts to diverge. This problem does not occur in the RB-SGMRES method and GCR, since the matrix U_n remains well-conditioned due to the low stagnation factor. These two implementations behave almost equally to the backward stable MGS-GMRES method. For numerical experiments with MGS-GMRES on the same examples, we refer the reader to [11] and [15].

4. Conclusions. In this paper we have studied the numerical behavior of several minimum residual methods mathematically equivalent to GMRES. Two general formulations have been analyzed: the generalized simpler approach that does not require an upper Hessenberg factorization and the generalized update approach which

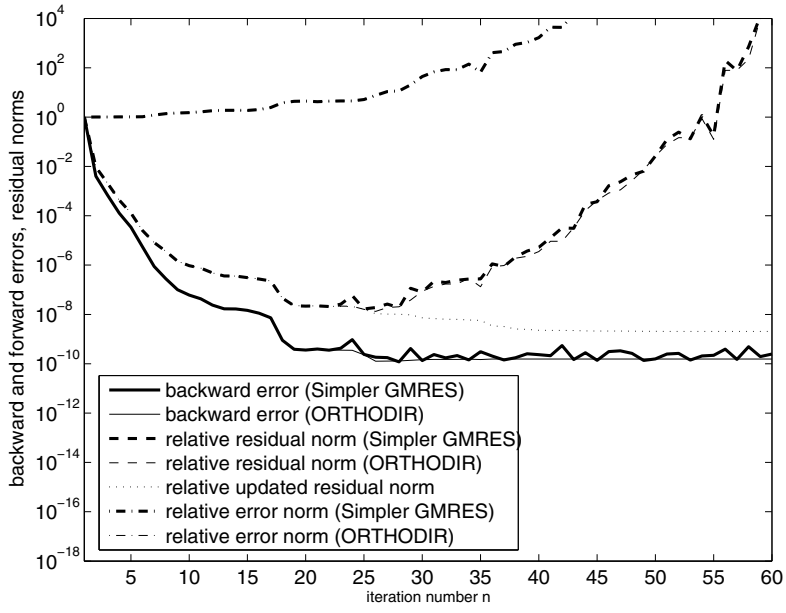


FIG. 3.1. *The test problem FS*1836 *solved by Simpler GMRES and ORTHODIR: Normwise backward error* $\|b - Ax_n\|/(\|A\|\|x_n\|)$ *(thick solid line: Simpler GMRES; thin solid line: ORTHODIR), relative true residual norm* $\|b - Ax_n\|/\|b\|$ *(thick dashed line: Simpler GMRES; thin dashed line: ORTHODIR), relative norm of the updated residual* $\|r_n\|/\|b\|$ *(dotted line), relative norms of the error* $\|x - x_n\|/\|x\|$ *(thick dash-dotted line: Simpler GMRES; thin dash-dotted line: ORTHODIR).*

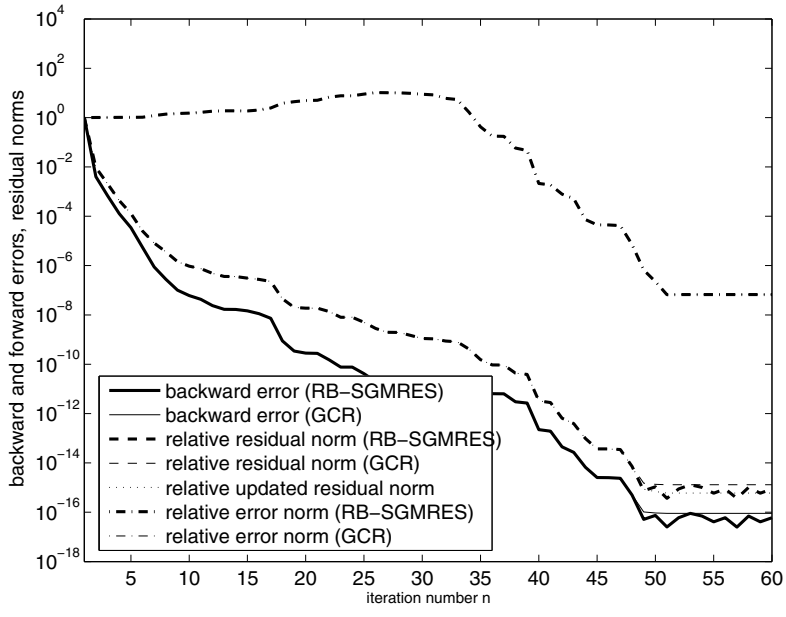


FIG. 3.2. *The test problem FS*1836 *solved by RB-SGMRES and GCR: Normwise backward error* $\|b - Ax_n\|/(\|A\|\|x_n\|)$ *(thick solid line: RB-SGMRES; thin solid line: GCR), relative true residual norm* $\|b - Ax_n\|/\|b\|$ *(thick dashed line: RB-SGMRES; thin dashed line: GCR), relative norm of the updated residual* $\|r_n\|/\|b\|$ *(dotted line), relative norms of the error* $\|x - x_n\|/\|x\|$ *(thick dash-dotted line: RB-SGMRES; thin dash-dotted line: GCR).*

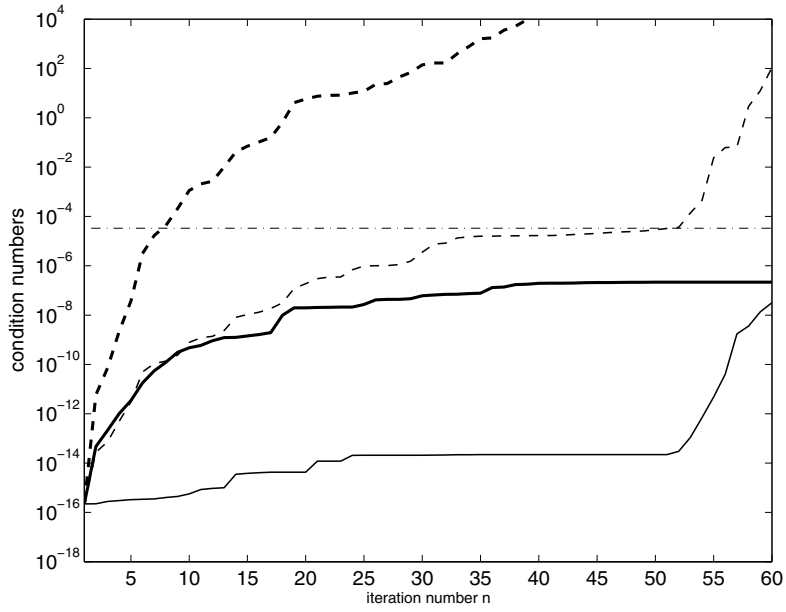


FIG. 3.3. *The test problem FS*1836*, condition numbers multiplied by unit roundoff u:* $u\kappa(A)$ *(dash-dotted line);* $u\kappa(Z_n)$ *(thick solid line) and* $u\kappa(U_n)$ *(thick dashed line) for* $Z_n = [\frac{r_0}{\|r_0\|}, V_{n-1}]$*;* $u\kappa(Z_n)$ *(thin solid line) and* $u\kappa(U_n)$ *(thin dashed line) for* $Z_n = \widetilde{R}_n$*.*

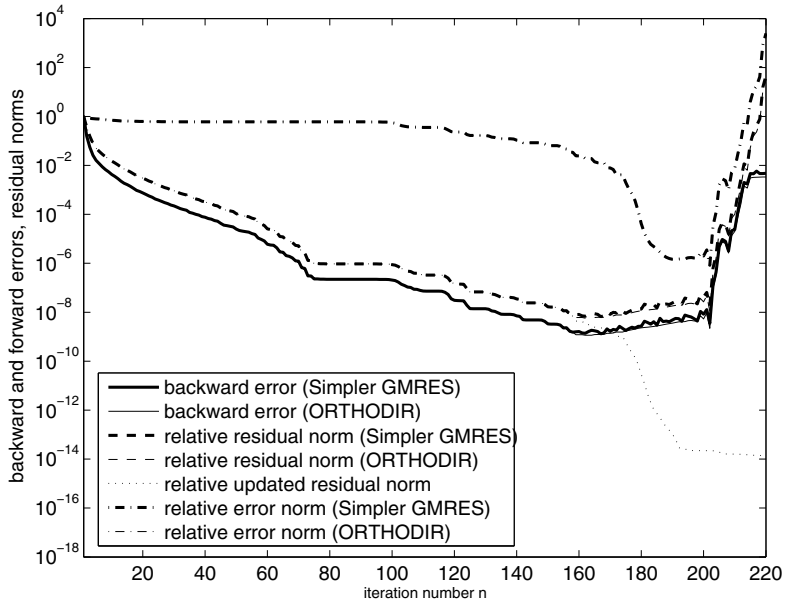


FIG. 3.4. *The test problem STEAM*1 *solved by Simpler GMRES and ORTHODIR: Normwise backward error* $\|b - Ax_n\|/(\|A\|\|x_n\|)$ *(thick solid line: Simpler GMRES; thin solid line: ORTHODIR), relative true residual norm* $\|b - Ax_n\|/\|b\|$ *(thick dashed line: Simpler GMRES; thin dashed line: ORTHODIR), relative norm of the updated residual* $\|r_n\|/\|b\|$ *(dotted line), relative norms of the error* $\|x - x_n\|/\|x\|$ *(thick dash-dotted line: Simpler GMRES; thin dash-dotted line: ORTHODIR).*

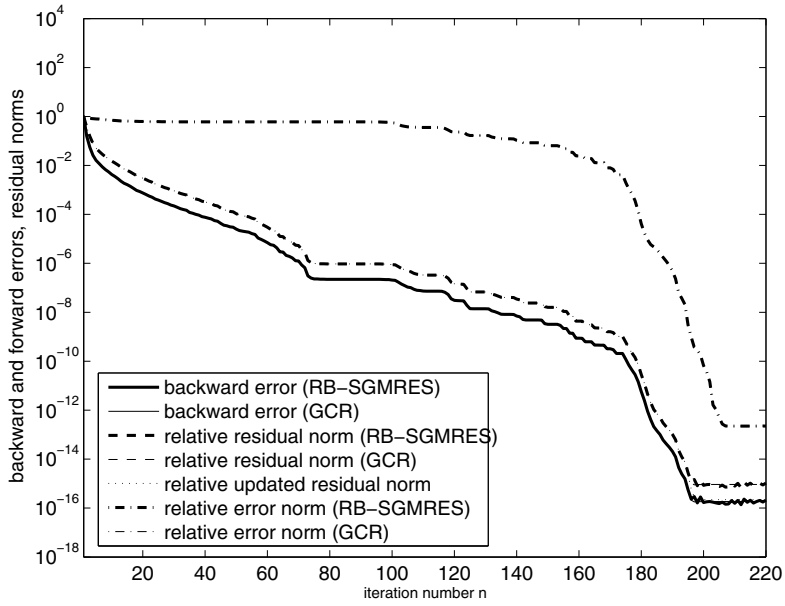


FIG. 3.5. *The test problem STEAM1 solved by RB-SGMRES and GCR: Normwise backward error* $\|b - Ax_n\|/(\|A\|\|x_n\|)$ *(thick solid line: RB-SGMRES; thin solid line: GCR), relative true residual norm* $\|b - Ax_n\|/\|b\|$ *(thick dashed line: RB-SGMRES; thin dashed line: GCR), relative norm of the updated residual* $\|r_n\|/\|b\|$ *(dotted line), relative norms of the error* $\|x - x_n\|/\|x\|$ *(thick dash-dotted line: RB-SGMRES; thin dash-dotted line: GCR).*

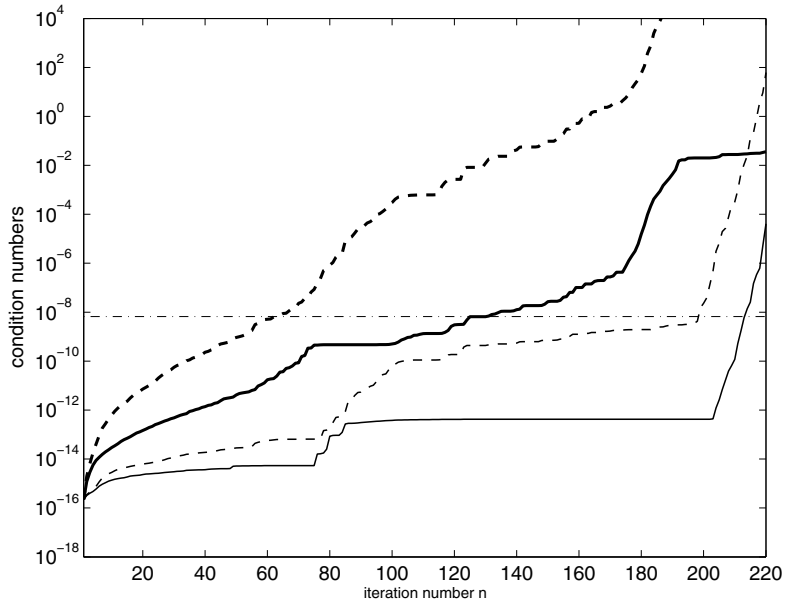


FIG. 3.6. *The test problem STEAM1, condition numbers multiplied by unit roundoff u:* $u\kappa(A)$ *(dash-dotted line);* $u\kappa(Z_n)$ *(thick solid line) and* $u\kappa(U_n)$ *(thick dashed line) for* $Z_n = [\frac{r_0}{\|r_0\|}, V_{n-1}]$*;* $u\kappa(Z_n)$ *(thin solid line) and* $u\kappa(U_n)$ *(thin dashed line) for* $Z_n = \widetilde{R}_n$*.*

is based on generating a sequence of appropriately computed direction vectors. It has been shown that for the generalized simpler approach our analysis leads to an upper bound for the backward error proportional to the roundoff unit, whereas for the generalized update approach the same quantity can be bounded by a term proportional to the condition number of A. Although our analysis suggests that the difference between both may be up to the order of $\kappa(A)$, in practice they behave very similarly, and it is very difficult to find a concrete example with a significant difference in the limiting accuracy measured by the normwise backward error of the approximate solutions x_n. Our first test problem displayed in Figure 2.1 is such a rare example. Moreover, when looking at the errors, we note that both approaches lead essentially to the same accuracy of x_n.

We have indicated that the choice of the basis Z_n is the most important issue for the stability of the considered schemes. Our analysis supports the well-known fact that, even when implemented with the best possible orthogonalization techniques, Simpler GMRES and ORTHODIR are inherently less stable due to the choice $Z_n = [\frac{r_0}{\|r_0\|}, V_{n-1}]$ for the basis. The situation becomes significantly better when we use the residual basis $Z_n = \widetilde{R}_n$. This choice leads to the popular GCR (ORTHOMIN, GMRESR) method, which is widely used in applications. Assuming some reasonable residual decrease (which happens almost always in finite precision arithmetic), we have shown that this scheme is quite efficient, and we have proposed a conditionally backward stable variant RB-SGMRES. Our theoretical results in a sense justify the use of the GCR method in practical computations. In this paper we studied only the unpreconditioned implementations. The implications for the preconditioned GCR scheme will be discussed elsewhere.

Acknowledgments. We would like to thank Julien Langou, Yvan Notay, Kees Vuik, and Gérard Meurant for valuable discussions during the preparation of the paper. We also thank the referees for their comments, which helped to improve the presentation of our results.

REFERENCES

[1] National Institute of Standards and Technology, *Matrix Market*, http://math.nist.gov/MatrixMarket.

[2] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[3] J. Drkošová, A. Greenbaum, M. Rozložník, and Z. Strakoš, *Numerical stability of GMRES*, BIT, 35 (1995), pp. 309–330.

[4] M. Eiermann and O. G. Ernst, *Geometric aspects of the theory of Krylov subspace methods*, Acta Numer., 10 (2001), pp. 251–312.

[5] S. C. Eisenstat, H. C. Elman, and M. H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 20 (1983), pp. 345–357.

[6] H. C. Elman, *Iterative Methods for Large Sparse Nonsymmetric Systems of Linear Equations*, Ph.D. thesis, Yale University, New Haven, CT, 1982.

[7] D. K. Faddeev and V. N. Faddeeva, *Computational Methods of Linear Algebra*, Fizmatgiz, Moskow, 1960 (in Russian).

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[9] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Frontiers Appl. Math. 17, SIAM, Philadelphia, 1997.

[10] A. Greenbaum and Z. Strakoš, *Matrices that generate the same Krylov residual spaces*, in Recent Advances in Iterative Methods, G. H. Golub, A. Greenbaum, and M. Luskin, eds., Springer-Verlag, New York, 1994, pp. 95–119.

[11] A. Greenbaum, M. Rozložník, and Z. Strakoš, *Numerical behaviour of the modified Gram-Schmidt GMRES implementation*, BIT, 37 (1997), pp. 706–719.
[12] M. H. Gutknecht and Z. Strakoš, *Accuracy of two three-term and three two-term recurrences for Krylov space solvers*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 213–229.
[13] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[14] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*,, Corrected reprint of the 1991 original, Cambridge University Press, Cambridge, UK, 1994.
[15] J. Liesen, M. Rozložník, and Z. Strakoš, *Least squares residuals and minimal residual methods*, SIAM J. Sci. Comput., 23 (2002), pp. 1503–1525.
[16] G. Meurant, *Computer Solution of Large Linear Systems*, Stud. Math. Appl. 28, North–Holland, Amsterdam, 1999.
[17] Y. Notay, *Personal communication*.
[18] C. C. Paige, M. Rozložník, and Z. Strakoš, *Modified Gram–Schmidt (MGS), least squares, and backward stability of MGS-GMRES*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 264–284.
[19] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
[20] M. Rozložník and Z. Strakoš, *Variants of residual minimizing Krylov subspace methods*, in Proceedings of the 6th Summer School Software and Algorithms of Numerical Mathematics, Ivo Marek, ed., University of West Bohemia, Pilsen, 1995, pp. 208–225.
[21] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
[22] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[23] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[24] V. Simoncini and D. B. Szyld, *New conditions for non-stagnation of minimal residual methods*, Numer. Math., 109 (2008), pp. 477–487.
[25] G. L. G. Sleijpen, H. A. van der Vorst, and J. Modersitzki, *Differences in the effects of rounding errors in Krylov solvers for symmetric indefinite linear systems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 726–751.
[26] J. van den Eshof, G. L. G. Sleijpen, and M. B. van Gijzen, *Iterative linear system solvers with approximate matrix-vector products*, in QCD and Numerical Analysis III, Lect. Notes Comput. Sci. Eng. 47, A. Borici, A. Frommer, B. Joo, A. D. Kennedy, and B. Pendleton, eds., Springer-Verlag, Berlin, 2005, pp. 133–142.
[27] A. van der Sluis, *Condition numbers and equilibration matrices*, Numer. Math., 14 (1969), pp. 14–23.
[28] H. A. van der Vorst and C. Vuik, *GMRESR: A family of nested GMRES methods*, Numer. Linear Algebra Appl., 1 (1994), pp. 369–386.
[29] P. K. W. Vinsome, *Orthomin, an iterative method for solving sparse sets of simultaneous linear equations*, in Proceedings of the Fourth Symposium on Reservoir Simulation, SPE of AIME, Los Angeles, CA, 1976, pp. 149–159.
[30] H. F. Walker and L. Zhou, *A simpler GMRES*, Numer. Linear Algebra Appl., 1 (1994), pp. 571–581.
[31] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice–Hall, Englewood Cliffs, NJ, 1963.
[32] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.
[33] D. M. Young and K. C. Jea, *Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods*, Linear Algebra Appl., 34 (1980), pp. 159–194.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1500–1519

ANALYSIS OF THE TRUNCATED SPIKE ALGORITHM*

CARL CHRISTIAN KJELGAARD MIKKELSEN† AND MURAT MANGUOGLU†

Abstract. The truncated SPIKE algorithm is a parallel solver for linear systems which are banded and strictly diagonally dominant by rows. There are machines for which the current implementation of the algorithm is faster and scales better than the corresponding solver in ScaLAPACK (PDDBTRF/PDDBTRS). In this paper we prove that the SPIKE matrix is strictly diagonally dominant by rows with a degree no less than the original matrix. We establish tight upper bounds on the decay rate of the spikes as well as the truncation error. We analyze the error of the method and present the results of some numerical experiments which show that the accuracy of the truncated SPIKE algorithm is comparable to LAPACK and ScaLAPACK.

Key words. direct methods, banded, and row diagonally dominant linear systems

AMS subject classification. 65F05

DOI. 10.1137/080719571

1. Introduction.

A matrix $A = [a_{ij}]$ is diagonally dominant by rows if

$$\sum_{i \neq j} |a_{ij}| \leq |a_{ii}| \tag{1.1}$$

for all i. If the inequality is sharp, then A is *strictly* diagonally dominant by rows.

The truncated SPIKE algorithm is a parallel solver for linear systems which are banded and strictly diagonally dominant by rows. Polizzi and Sameh demonstrated [10], [11] that there are parallel machines for which the algorithm is faster and scales better than the algorithm which is implemented in ScaLAPACK (PDDBTRF/PDDBTRS) [1]. We present the algorithm in section 2 and prove certain key properties of the truncated SPIKE algorithm in section 3. We analyze the error in section 4. We present the results of some experiments which supplement our theoretical analysis, and we compare the accuracy of the truncated SPIKE algorithm and ScaLAPACK in section 5.

The SPIKE algorithms are designed to solve banded systems on a parallel machine. The basic idea was introduced by Sameh and Kuck [12] who considered the tridiagonal case and Chen, Kuck, and Sameh [2] who studied the triangular case. Lawrie and Sameh [8] applied the algorithm to the symmetric positive definite systems, while Dongarra and Sameh [4] considered the strictly diagonally dominant case. Variations of the SPIKE algorithms for tridiagonal systems were introduced by Sun, Zhang, and Ni [13], who also analyzed the truncation error for tridiagonal systems which are evenly diagonally dominant. The truncation error for tridiagonal Toeplitz systems, which are also strictly diagonally dominant, as well as symmetric or skew symmetric was considered by Sun [14]. Another variation of the SPIKE algorithm for strictly diagonally dominant systems was studied by Larriba-Pey, Jorba, and Navarro

*Received by the editors March 31, 2008; accepted for publication (in revised form) by R.-C. Li August 25, 2008; published electronically December 3, 2008. This research is supported by the National Science Foundation (NSF-CCF-0635169), the Air Force Research Laboratory (FA8750-06-1-0233), and the Intel Corporation.

http://www.siam.org/journals/simax/30-4/71957.html

†Department of Computer Science, Purdue University, West Lafayette, IN 47907 (cmikkels@cs.purdue.edu, mmanguog@cs.purdue.edu).

[7]. Polizzi and Sameh have extended the SPIKE algorithms to the general banded case, and they developed the SPIKE package.

If A is nonsingular and diagonally dominant by rows, then the diagonal entries are nonzero and the dominance factor [5] ϵ is defined as follows:

$$\epsilon = \max_i \left\{ \frac{\sum_{i \neq j} |a_{ij}|}{|a_{ii}|} \right\}. \tag{1.2}$$

If $\epsilon > 0$, then the degree of diagonal dominance d is given by

$$d = \epsilon^{-1}. \tag{1.3}$$

The degree of diagonal dominance is central to the analysis of the truncated SPIKE algorithm.

2. The algorithm. Consider the nonsingular linear system

$$Ax = f,$$

where A is a n by n banded matrix which is strictly diagonally dominant by rows.

We assume that the number of superdiagonals k is equal to the number of subdiagonals and that the matrix is narrow banded, i.e., $k \ll n$. Let p denote the number of processors. We assume for simplicity that p divides n. Let the system be partitioned into the block diagonal form shown below

$$Ax = \begin{bmatrix} A_1 & \overline{B}_1 & & & \\ \overline{C}_2 & A_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \overline{B}_{p-1} \\ & & & \overline{C}_p & A_p \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_p \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_p \end{bmatrix}, \tag{2.1}$$

where $A_i, i = 1, 2, \ldots, p$ is a banded matrix of order $\mu = n/p$ and bandwidth $2k + 1$ (just like A),

$$\overline{B}_i = \begin{bmatrix} 0 & 0 \\ B_i & 0 \end{bmatrix}, \quad \text{and} \quad \overline{C}_{i+1} = \begin{bmatrix} 0 & C_{i+1} \\ 0 & 0 \end{bmatrix}, \quad i = 1, 2, \ldots p-1,$$

in which B_i and C_i are lower and upper triangular matrices, respectively, each of order k. Let D denote the main block diagonal D, i.e.,

$$D = \text{diag}\{A_1, A_2, \ldots, A_p\}.$$

The matrix D is nonsingular because A is strictly diagonally dominant. If we premultiply both sides of (2.1) by D^{-1}, we obtain a system $Sx = g$ of the form

$$\begin{bmatrix} I_\mu & \overline{V}_1 & & & \\ \overline{W}_2 & I_\mu & \overline{V}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \overline{W}_{p-1} & I_\mu & \overline{V}_{p-1} \\ & & & \overline{W}_p & I_\mu \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{p-1} \\ x_p \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{p-1} \\ g_p \end{bmatrix}, \tag{2.2}$$

where

$$\overline{V}_i = \begin{bmatrix} V_i & 0 \end{bmatrix}, \quad \text{and} \quad \overline{W}_i = \begin{bmatrix} 0 & W_i \end{bmatrix},$$

in which V_i and W_i are matrices with k columns given by

$$V_i = A_i^{-1} \begin{bmatrix} 0 \\ B_i \end{bmatrix}, \quad \text{and} \quad W_i = A_i^{-1} \begin{bmatrix} C_i \\ 0 \end{bmatrix},$$

and will in general be full. Let the tall and narrow matrices V_i and W_i be partitioned, in turn, as follows

$$V_i = \begin{bmatrix} V_i^{(t)} \\ V_i^{(m)} \\ V_i^{(b)} \end{bmatrix}, \quad \text{and} \quad W_i = \begin{bmatrix} W_i^{(t)} \\ W_i^{(m)} \\ W_i^{(b)} \end{bmatrix},$$

where $V_i^{(t)}$, $V_i^{(b)}$, $W_i^{(t)}$, and $W_i^{(b)} \in \mathbb{R}^{k \times k}$. The superscripts t, m, and b are abbreviations of the words top, middle, and bottom, respectively. Let x_i and g_i be partitioned conformally, i.e.,

$$x_i = \begin{bmatrix} x_i^{(t)} \\ x_i^{(m)} \\ x_i^{(b)} \end{bmatrix}, \quad \text{and} \quad g_i = \begin{bmatrix} g_i^{(t)} \\ g_i^{(m)} \\ g_i^{(b)} \end{bmatrix}.$$

As an illustration we show the system (2.2) for $p = 3$, where $\nu = \mu - 2k$,

$$\begin{bmatrix} I_k & & & V_1^{(t)} & & & & & \\ & I_\nu & & V_1^{(m)} & & & & & \\ & & I_k & V_1^{(b)} & & & & & \\ & & W_2^{(t)} & I_k & & & V_2^{(t)} & & \\ & & W_2^{(m)} & & I_\nu & & V_2^{(m)} & & \\ & & W_2^{(b)} & & & I_k & V_2^{(b)} & & \\ & & & & & W_3^{(t)} & I_k & & \\ & & & & & W_3^{(t)} & & I_\nu & \\ & & & & & W_3^{(t)} & & & I_k \end{bmatrix} \begin{bmatrix} x_1^{(t)} \\ x_1^{(m)} \\ x_1^{(b)} \\ x_2^{(t)} \\ x_2^{(m)} \\ x_2^{(b)} \\ x_3^{(t)} \\ x_3^{(m)} \\ x_3^{(b)} \end{bmatrix} = \begin{bmatrix} g_1^{(t)} \\ g_1^{(m)} \\ g_1^{(b)} \\ g_2^{(t)} \\ g_2^{(m)} \\ g_2^{(b)} \\ g_3^{(t)} \\ g_3^{(m)} \\ g_3^{(b)} \end{bmatrix}.$$

It is from these narrow block columns or spikes extending from the main diagonal that the algorithm has derived its name. We will frequently refer to the V_i as the superdiagonal spikes and to the W_i as the subdiagonal spikes, and call the matrix S the SPIKE matrix.

Observe that the union of the k equations above and the k equations below the $p-1$ partition lines forms an independent subsystem of order $2k(p-1)$. We shall refer to this system as the "reduced" system and write $Rx_r = g_r$. It has the form

$$\begin{bmatrix} E_1 & F_1 & & & \\ G_2 & E_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & F_{p-2} \\ & & & G_{p-2} & E_{p-1} \end{bmatrix} \begin{bmatrix} x_{r,1} \\ x_{r,2} \\ \vdots \\ x_{r,p-1} \end{bmatrix} = \begin{bmatrix} g_{r,1} \\ g_{r,2} \\ \vdots \\ g_{r,p-1} \end{bmatrix}, \tag{2.3}$$

where

$$E_i = \begin{bmatrix} I_k & V_i^{(b)} \\ W_{i+1}^{(t)} & I_k \end{bmatrix}, \quad F_i = \begin{bmatrix} 0 & 0 \\ 0 & V_{i+1}^{(t)} \end{bmatrix}, \quad \text{and} \quad G_i = \begin{bmatrix} W_i^{(b)} & 0 \\ 0 & 0 \end{bmatrix},$$

and

$$x_{r,i} = \begin{bmatrix} x_i^{(b)} \\ x_{i+1}^{(t)} \end{bmatrix}, \quad \text{and} \quad g_{r,i} = \begin{bmatrix} g_i^{(b)} \\ g_{i+1}^{(t)} \end{bmatrix}.$$

The subscript r is an abbreviation of the word "reduced". Dongarra and Sameh [4] noted that the reduced system is strictly diagonally dominant by rows and solved the reduced system using a parallel implementation of the Jacobi iteration. In Theorem 3.3 we show that the reduced system is strictly diagonally dominant by rows with a degree no less than the original matrix.

Once the reduced system has been solved

$$z_i = g_i - W_i x_{i-1}^{(b)} - V_i x_{i+1}^{(t)},$$

where x_0, x_{p+1}, W_1, and V_p are undefined and should be taken to zero in this equation. If the calculations are carried out using exact arithmetic, then z is the solution of $Ax = f$.

In general the reduced system is block tridiagonal. However, Polizzi and Sameh [10] noted that the off diagonal blocks are often negligible and can be dropped, yielding a truncated reduced system $Tx_{tr} = g_r$, which is block diagonal,

$$\begin{bmatrix} E_1 & & & & \\ & E_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & E_{p-1} \end{bmatrix} \begin{bmatrix} x_{tr,1} \\ x_{tr,2} \\ \vdots \\ x_{tr,p-1} \end{bmatrix} = \begin{bmatrix} g_{r,1} \\ g_{r,2} \\ \vdots \\ g_{r,p-1} \end{bmatrix}. \tag{2.4}$$

The subscript tr is an abbreviation of the words "truncated" and "reduced". In Theorem 3.8 we establish a tight upper bound on the size of the off diagonal blocks in terms of the degree of diagonal dominance of the original matrix and the size of the partitions. Polizzi and Sameh [10] showed that it is possible to compute the truncated reduced system without assembling the entire SPIKE system. Let $\mathcal{A}$ denote one of the diagonal blocks and consider the problem of computing the bottom $\mathcal{V}^{(b)}$ of the corresponding spike $\mathcal{V}$, given by

$$\mathcal{A}\mathcal{V} = \begin{bmatrix} 0 \\ \mathcal{B} \end{bmatrix}, \tag{2.5}$$

where $\mathcal{B}$ is a k by k dense matrix. It is not important here that $\mathcal{B}$ is lower triangular. We can exploit the remaining structure as follows. Let $\mathcal{A} = LU$ be the LU factorization of $\mathcal{A}$. Partition L and Y conformally with the right-hand side,

$$\begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathcal{B} \end{bmatrix},$$

where L_{22} is a k by k lower unit triangular matrix. Since $L_{11}Y_1 = 0$, we have $Y_1 = 0$, and the problem reduces to solving $L_{22}Y_2 = \mathcal{B}$. Then we solve $U\mathcal{V} = Y$. Partition U and $\mathcal{V}$ conformally with Y and the right-hand side,

$$\begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \begin{bmatrix} \mathcal{V}_1 \\ \mathcal{V}_2 \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}.$$

Since U is upper triangular we can compute $\mathcal{V}^{(b)} = \mathcal{V}_2 = U_{22}^{-1}Y_2$ without computing $\mathcal{V}_1$. Similarly if $A_i = U_i'L_i'$ is a UL factorization of A_i, then it is possible to extract the top of the subdiagonal spikes without computing the entire spike.

Polizzi and Sameh [10] found experimentally that it is faster to extract the truncated reduced system using the LU/UL combinations than it is to compute the entire SPIKE matrix using the LU factorizations only. This is true on machines where arithmetic operations require much less time than memory references. The UL/LU strategy has greater data locality: computing the spikes is a BLAS2 operation, whereas computing the LU/UL factorizations is a BLAS3 operation.

The original equation is equivalent to

$$A_i x_i = f_i - C_i x_{i-1}^{(b)} - B_i x_{i+1}^{(t)}, \quad i = 1, 2, \ldots p, \tag{2.6}$$

where $x_0^{(b)}$, $x_{p+1}^{(t)}$, C_1, and B_p are undefined and should be taken to zero in this equation.

These observations led to the truncated SPIKE algorithm by Polizzi and Sameh [10]. The algorithm consist of four stages.

Stage 1. Processor i computes the LU/UL factorizations

$$A_i = L_i U_i \quad \text{and} \quad A_i = U_i' L_i', \quad i = 1, 2, \ldots p.$$

Stage 2. Processor i solves

$$A_i g_i = f_i, \quad i = 1, 2, \ldots p,$$

using the LU factorization. Processor i computes $V_i^{(b)}$ using (L_i, U_i), $i = 1, 2, \ldots p-1$. Processor i computes $W_i^{(t)}$ using (U_i', L_i') , $i = 2, 3 \ldots p$.

Stage 3. Processor $i+1$ sends $W_{i+1}^{(t)}$ and $g_{i+1}^{(t)}$ to processor i, $i = 1, 2, \ldots p-1$. Processor i solves one block of the truncated reduced system, specifically

$$\begin{bmatrix} I_k & V_i^{(b)} \\ W_{i+1}^{(t)} & I_k \end{bmatrix} \begin{bmatrix} x_i^{(b)} \\ x_{i+1}^{(t)} \end{bmatrix} = \begin{bmatrix} g_i^{(b)} \\ g_{i+1}^{(t)} \end{bmatrix}, \quad i = 1, 2, \ldots p-1,$$

using Gaussian elimination without pivoting.

Stage 4. Processor i sends $x_i^{(b)}$ to processor $i+1$, for $i = 1, 2, \ldots, p-1$, and processor i sends $x_i^{(t)}$ to processor $i-1$ for $i = 2, 3, \ldots, p$. Then processor i solves

$$A_i y_i = f_i - C_i x_{i-1}^{(b)} - B_i x_{i+1}^{(t)}, \quad i = 1, 2, \ldots, p$$

using the LU factorization, where $x_0^{(b)}$, $x_{p+1}^{(t)}$, C_1, and B_p are undefined and should be taken to zero in this equation. The vector y is an approximation of the solution to $Ax = f$.

3. The matrices S, R, and T. In this section we prove that the matrices S, R, and T in (2.2), (2.3), and (2.4) are strictly diagonally dominant by rows with degree no less than A, and we establish an upper bound on their condition number. The degree of diagonal dominance is defined by (1.3). We bound the truncation error, i.e., the difference between R and T, and show that all our bounds are tight.

The general estimates for the decay rates of the inverse of a banded matrix discovered by Demko, Moss, and Smith [3] are not suitable in our situation because it is necessary to exploit the relationship between the matrices and the right-hand sides which determine the spikes, in order to obtain estimates which are tight.

LEMMA 3.1. *Let $n \le m$ and let A be any n by m matrix which is strictly diagonally dominant by rows with degree $d > 1$. Let $A = LU$ be the LU factorization which is obtained by applying Gaussian elimination without pivoting to A. Then U is strictly diagonally dominant by rows with degree no less than d.*

Proof. Gaussian elimination produces a chain of matrices $A^{(j)}$, where the first $j-1$ columns of $A^{(j)}$ are lower triangular, $A = A^{(1)}$ and $A^{(n)} = U$. Due to the recursive nature of Gaussian elimination, it suffices to consider the transition from $A = A^{(1)}$ to $B = A^{(2)}$. Let $B = [b_{ij}]$. We must show the following equalities

$$|b_{kk}| \ge d \sum_{j \notin \{1,k\}} |b_{k,j}|, \quad k = 2, 3, \ldots, m.$$

Now, since $d \ge 1$ and $|a_{11}| \ge d \sum_{j=2}^{m} |a_{1j}|$ we have

$$\begin{aligned} |a_{kk}| &\ge d \sum_{j \ne k} |a_{kj}| \ge |a_{k1}| + d \sum_{j \notin \{1,k\}} |a_{kj}| \\ &\ge |a_{k1}| \frac{d \sum_{j=2}^{m} |a_{1j}|}{|a_{11}|} + d \sum_{j \notin \{1,k\}} |a_{kj}| \\ &\ge |a_{k1}| \frac{|a_{1k}|}{|a_{11}|} + d \sum_{j \notin \{1,k\}} \left(|a_{kj}| + \frac{|a_{k1}|}{|a_{11}|} |a_{1j}| \right). \end{aligned}$$

Now, since

$$b_{ij} = a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j}, \quad i = 2, 3, \ldots, n, \quad j = 2, 3, \ldots, m,$$

the previous inequality implies

$$|b_{kk}| \ge |a_{kk}| - \frac{|a_{k1}|}{|a_{11}|} |a_{1k}| \ge d \sum_{j \notin \{1,k\}} \left(|a_{kj}| + \frac{|a_{k1}|}{|a_{11}|} |a_{1j}| \right) \ge d \sum_{j \notin \{1,k\}} |b_{kj}|. \qquad \square$$

COROLLARY 3.2. *Let A be an n by n matrix, and let F be an n by m matrix. If the matrix $\begin{bmatrix} A, & F \end{bmatrix}$ is strictly diagonally dominant by rows with degree $d > 1$, then the matrix $\begin{bmatrix} I, & A^{-1}F \end{bmatrix}$ is strictly diagonally dominant by rows with degree no less than d.*

Proof. We use Gaussian elimination without pivoting to reduce the n by $n+m$ matrix $\begin{bmatrix} A, & F \end{bmatrix}$ to upper triangular form, $U = [u_{ij}]$. By Lemma 3.1, U is diagonally dominant by rows with degree no less than d, and using back substitution we have a formula for the entries g_{ij} of the n by m matrix $G = A^{-1}F$, namely,

$$g_{n-t,j} = \frac{1}{u_{n-t,n-t}} \left(u_{n-t,n+j} - \sum_{s=n-t+1}^{n} u_{n-t,s} g_{s,j} \right),$$

for $j = 1, 2, \dots m$ and $t = 0, 1, 2 \dots n-1$. Let $\epsilon = d^{-1}$ and let $\Omega \subseteq \{0, 1, \dots n-1\}$ be given by

$$t \in \Omega \Leftrightarrow \sum_{j=1}^{m} |g_{n-t,j}| \leq \epsilon.$$

We will prove that $\Omega = \{0, 1, \dots n-1\}$. First, $0 \in \Omega$ because U is strictly diagonally dominant by rows with degree no less than d, and if $\{0, 1, 2, \dots t-1\} \subset \Omega$ with $t < n$, then

$$\begin{aligned}
\sum_{j=1}^{m} |g_{n-t,j}| &\leq \frac{1}{|u_{n-t,n-t}|} \sum_{j=1}^{m} \left(|u_{n-t,n+j}| + \sum_{s=n-t+1}^{n} |u_{n-t,s}||g_{s,j}| \right) \\
&= \frac{1}{|u_{n-t,n-t}|} \left(\sum_{j=1}^{m} |u_{n-t,n+j}| + \sum_{s=n-t+1}^{n} |u_{n-t,s}| \sum_{j=1}^{m} |g_{s,j}| \right) \\
&\leq \frac{1}{|u_{n-t,n-t}|} \left(\sum_{j=1}^{m} |u_{n-t,n+j}| + \sum_{s=n-t+1}^{n} |u_{n-t,s}|\epsilon \right) \leq \epsilon,
\end{aligned}$$

which implies $t \in \Omega$. Therefore $\Omega = \{0, 1, 2 \dots n-1\}$ and the proof is complete. ∎

THEOREM 3.3. *Let A be strictly diagonally dominant by rows with degree $d > 1$. Then the matrices S, R, and T are strictly diagonally dominant by rows with degree no less than d, specifically*

$$d \leq d(S) \leq d(R) \leq d(T),$$

with equality possible. The condition numbers share a common bound, namely

$$\max\{\, \kappa_\infty(S),\, \kappa_\infty(R),\, \kappa_\infty(T)\} \leq \frac{d+1}{d-1},$$

with the possibility of

$$\kappa_\infty(S) = \kappa_\infty(R) = \kappa_\infty(T) = \frac{d+1}{d-1}.$$

Proof. If S is strictly diagonally dominant by rows, then it is clear that T and R are strictly diagonally dominant by rows and $d(S) \leq d(R) \leq d(T)$. By applying Lemma 3.2 to the matrices $\begin{bmatrix} A_i, & F_i \end{bmatrix}$ where

$$F_1 = \begin{bmatrix} 0 \\ B_1 \end{bmatrix}, \quad F_i = \left[\begin{bmatrix} 0 \\ B_i \end{bmatrix}, \begin{bmatrix} C_i \\ 0 \end{bmatrix} \right], \; i = 2, \dots p-1, \quad \text{and} \quad F_p = \begin{bmatrix} C_p \\ 0 \end{bmatrix},$$

we see that S is strictly diagonally dominant by rows with degree no less than d. Since $S_{ii} = 1$, we have $\|S - I\|_\infty \leq \epsilon < 1$ which allows us to treat S as a small perturbation of the identity matrix and estimate

$$\|S^{-1}\|_\infty \leq \frac{1}{1-\epsilon}, \quad \text{and} \quad \kappa_\infty(S) \leq \frac{1+\epsilon}{1-\epsilon} = \frac{d+1}{d-1},$$

and similarly for R and T.

It remains to be seen that our bounds are tight. To this end we consider a special case of the original problem, (2.1), where the diagonal blocks satisfy $A_i = I_\mu$ and the off-diagonal blocks are given by

$$\overline{B}_i = \begin{bmatrix} 0 & 0 \\ \epsilon J_k & 0 \end{bmatrix}, \quad \text{and} \quad \overline{C}_{i+1} = \begin{bmatrix} 0 & \epsilon J_k \\ 0 & 0 \end{bmatrix}, \quad i = 1, 2, \ldots p-1,$$

where J_k is the k by k antidiagonal identity matrix, and $\epsilon \in (0, 1)$. The matrix A is diagonally dominant by rows with degree $d = \epsilon^{-1}$. The upper and the lower bandwidths are equal to k. The main block diagonal is equal to the identity matrix, which implies $A = S$. The reduced system is block diagonal which implies $T = R$. It follows that

$$d(T) = d(R) = d(S) = d.$$

Computing S^{-1} reduces to inverting the 2 by 2 matrix $\left[\begin{smallmatrix} 1 & \epsilon \\ \epsilon & 1 \end{smallmatrix}\right]$. Direct computation establishes that

$$\kappa_\infty(T) = \kappa_\infty(R) = \kappa_\infty(S) = \frac{1+\epsilon}{1-\epsilon} = \frac{d+1}{d-1}. \qquad \square$$

We now study the truncation error, i.e., $\|R - T\|_\infty$. Let $\mathcal{A}$ denote one of the diagonal blocks of A, and let $\mathcal{V}$ be the corresponding superdiagonal spike given by (2.5). We are especially interested in the size of the elements at the top of the spike, i.e., the submatrix $\mathcal{V}^{(t)}$, which is given by

$$\mathcal{V}^{(t)} = \mathcal{V}(1:k, 1:k).$$

There is no loss of generality in limiting the analysis to the first diagonal block, rather there is a slight notational advantage, because the numbering of the elements of A and $\mathcal{A}$ coincide. We will use μ to denote the size of the first diagonal block.

We begin by estimating the size of the elements located in the bottom of $\mathcal{V}$; i.e., the submatrix $\mathcal{V}^{(b)}$ given by

$$\mathcal{V}^{(b)} = \mathcal{V}(\mu - k + 1 : \mu, 1 : k).$$

LEMMA 3.4. *Let A be strictly diagonally dominant by rows with degree $d > 1$. Let $\mathcal{V}$ be a superdiagonal spike. Then the submatrix $\mathcal{V}^{(b)}$ satisfies*

$$\|\mathcal{V}^{(b)}\|_\infty \leq \epsilon,$$

where $\epsilon = d^{-1}$.

Proof. Reduce the first μ by n block row to upper triangular form U. Since Gaussian elimination without pivoting preserves the upper bandwidth and does not decrease the degree of diagonal dominance, we have the following set of inequalities

$$\sum_{j=1}^{k} |u_{\mu-t,\mu-t+j}| \leq \epsilon |u_{\mu-t,\mu-t}|, \qquad t = 0, 1, \ldots k-1. \tag{3.1}$$

Our goal is to show that $\|\mathcal{V}^{(b)}\|_\infty \leq \epsilon$ or equivalently

$$\sum_{j=1}^{k} |v_{\mu-t,j}| \leq \epsilon, \quad t = 0, 1, \ldots k-1. \tag{3.2}$$

To this end we define the set $\Omega \subseteq \{0, 1, 2, \ldots, k-1\}$ by

$$t \in \Omega \Leftrightarrow \sum_{j=1}^{k} |v_{n-t,j}| \leq \epsilon.$$

We claim that $\Omega = \{0, 1, 2, \ldots, k-1\}$. Clearly $0 \in \Omega$, because

$$\sum_{j=1}^{k} |u_{\mu,\mu+j}| \leq \epsilon |u_{\mu,\mu}|, \quad \text{and} \quad v_{\mu,j} = \frac{u_{\mu,\mu+j}}{u_{\mu,\mu}}, \quad j = 1, 2, \ldots k.$$

Now suppose $\{0, 1, 2, \ldots t-1\} \subset \Omega$ with $t < k$. We wish to show that $t \in \Omega$. By back substitution we find that

$$v_{\mu-t,j} = \frac{1}{u_{\mu-t,\mu-t}} \left(u_{\mu-t,\mu+j} - \sum_{s=1}^{t} u_{\mu-t,\mu-t+s} v_{\mu-t+s,j} \right), \quad j = 1, 2, \ldots, (k-t),$$

and

$$v_{\mu-t,j} = -\frac{1}{u_{\mu-t,\mu-t}} \sum_{s=1}^{t} u_{\mu-t,\mu-t+s} v_{\mu-t+s,j}, \quad j = (k-t)+1, \ldots, k.$$

It follows that

$$\begin{aligned}
\sum_{j=1}^{k} |v_{\mu-t,j}| &\leq \frac{1}{|u_{\mu-t,\mu-t}|} \left(\sum_{j=1}^{k-t} |u_{\mu-t,\mu+j}| + \sum_{j=1}^{k} \sum_{s=1}^{t} |u_{\mu-t,\mu-t+s} v_{\mu-t+s,j}| \right) \\
&= \frac{1}{|u_{\mu-t,\mu-t}|} \left(\sum_{j=1}^{k-t} |u_{\mu-t,\mu+j}| + \sum_{s=1}^{t} |u_{\mu-t,\mu-t+s}| \sum_{j=1}^{k} |v_{\mu-t+s,j}| \right) \\
&\leq \frac{1}{|u_{\mu-t,\mu-t}|} \left(\sum_{j=1}^{k-t} |u_{\mu-t,\mu+j}| + \epsilon \sum_{s=1}^{t} |u_{\mu-t,\mu-t+s}| \right) \\
&\leq \frac{1}{|u_{\mu-t,\mu-t}|} \sum_{j=1}^{k} |u_{\mu-t,\mu-t+j}| \leq \epsilon,
\end{aligned}$$

which implies $t \in \Omega$. It follows that $\Omega = \{0, 1, 2, \ldots k-1\}$ and $\|\mathcal{V}^{(b)}\|_\infty \leq \epsilon$. ∎

We continue with the following lemma which relates the size of the elements in a specific row of $\mathcal{V}$ to the infinity norm of the k by k submatrix which lies directly below the row.

LEMMA 3.5. *Let μ denote the dimension of the diagonal block $\mathcal{A}$ and let $i \geq \mu - k$. Then*

$$\sum_{j=1}^{k} |v_{i,j}| \leq \epsilon \|\mathcal{V}(i+1 : i+k, 1 : k)\|_\infty.$$

Proof. We have

$$\mathcal{V} = \mathcal{A}^{-1} \begin{bmatrix} 0 \\ \mathcal{B} \end{bmatrix}$$

for the appropriate k by k matrix $\mathcal{B}$. We use Gaussian elimination without pivoting to reduce the matrix

$$\begin{bmatrix} \mathcal{A}, & \begin{bmatrix} 0 \\ \mathcal{B} \end{bmatrix} \end{bmatrix},$$

to upper triangular form $U = [u_{ij}]$. By Lemma 3.2 U is strictly diagonally dominant by rows with degree no less than d. Since the original matrix A was banded and no pivoting was applied, it follows that $u_{ij} = 0$ for all i and j such that $j > \mu$ and $i \geq \mu - k$. It follows by back substitution that

$$v_{i,j} = -\frac{1}{u_{i,i}} \sum_{s=i+1}^{i+k} u_{i,s} v_{s,j},$$

which implies

$$\sum_{j=1}^{k} |v_{i,j}| \leq \frac{1}{|u_{i,i}|} \sum_{s=i+1}^{i+k} |u_{i,s}| \sum_{j=1}^{k} |v_{s,j}|.$$

By definition

$$\max_{s=i+1,\ldots i+k} \sum_{j=1}^{k} |v_{s,j}| = \|\mathcal{V}(i+1:i+k, 1:k)\|_\infty,$$

and since U is strictly diagonally dominant by rows with degree no less than d, we have

$$\frac{1}{|u_{i,i}|} \sum_{s=i+1}^{i+k} |u_{i,s}| \leq \epsilon,$$

which completes the proof. □

The following corollary is an immediate consequence.

COROLLARY 3.6. *Let $\mathcal{V}'$ and $\mathcal{V}''$ be two k by k submatrices of the superdiagonal spike $\mathcal{V}$, such that $\mathcal{V}'$ lies directly on top of $\mathcal{V}''$. Then*

$$\|\mathcal{V}'\|_\infty \leq \epsilon \|\mathcal{V}''\|_\infty.$$

This corollary establishes a chain of inequalities leading from the bottom to the top of the spike which together with Lemma 3.4 implies the following theorem.

THEOREM 3.7. *Let d denote the degree of diagonal dominance of A, let μ denote the dimension of one of the diagonal blocks, and $q = \lfloor \mu/k \rfloor$ is the largest integer less than or equal to μ/k. The top of the corresponding superdiagonal spike $\mathcal{V}$ satisfies the inequality*

$$\|\mathcal{V}^{(t)}\|_\infty \leq \epsilon^q.$$

Is this estimate for the decay rate of the spikes tight or not? Let $\epsilon \in (0,1)$ and consider the upper triangular matrix A given by $a_{ii} = 1$, $a_{ij} = \epsilon$ for $i = j - k$, and $a_{ij} = 0$ in all other cases. Now consider a partition of a certain size μ. Write $\mu = qk + r$, where $q = \lfloor \mu/k \rfloor$, and the remainder r satisfies $0 \leq r < k$. If $r > 0$, then by back substitution we find that the corresponding spike is given by

$$\mathcal{V} = \begin{bmatrix} \mathcal{V}_{q+1}^T & \mathcal{V}_q^T & \ldots & \mathcal{V}_1^T \end{bmatrix}^T,$$

where

$$V_j = (-1)^{j-1}\epsilon^j I_k \quad \text{for } j = 1, 2, \ldots, q,$$

and $V_{q+1} = (-1)^q \epsilon^{q+1} E_r$, where I_k is the k by k identity matrix and E_r consists of the last r rows of I_k. If $r = 0$, then the term V_{q+1} does not appear. Regardless of the value of the remainder r, we have

$$\left\|\mathcal{V}^{(t)}\right\|_\infty = \epsilon^q.$$

In short, if we limit ourselves to matrices A which are strictly diagonally dominant by rows with degree d and upper bandwidth k, then the estimate given in Theorem 3.7 is tight.

The following theorem is an immediate consequence of Theorem 3.7.

THEOREM 3.8. *Let A be a n by n narrow banded matrix with upper and lower bandwidth k, and strictly diagonally dominant by rows with degree d. Then the truncation error satisfies*

$$\|R - T\|_\infty \leq \max_{i=1,\ldots p} d^{-q_i},$$

where $q_i = \lfloor \mu_i / k \rfloor$, and μ_i is the size of the ith partition.

A better bound exists in the special case in which A is a tridiagonal, evenly diagonally dominant matrix [13], or when A is a tridiagonal Toeplitz matrix, which is also strictly diagonally dominant, as well as symmetric or skew symmetric [14].

Now, consider for the sake of simplicity, the case when the partitions have the same size μ. Then Theorem 3.8 reduces to the statement

$$\|R - T\|_\infty \leq d^{-q},$$

where $q = \lfloor \mu / k \rfloor$. Let S_T denote the matrix obtained by eliminating the tips of the spikes from the spike matrix S. Then the reduced system matrix for S_T is equal to T. The truncation error effectively replaces A with the matrix $A_T = DS_T$, for which we have

$$\|A - A_T\|_\infty \leq \|D\|_\infty \|S - S_T\|_\infty \leq \|A\|_\infty \|R - T\|_\infty \leq d^{-q}\|A\|_\infty, \tag{3.3}$$

or equivalently $A_T = A + \Delta A$, where $\|\Delta A\|_\infty \leq d^{-q}\|A\|_\infty$. We see that the effect of the truncation is to introduce a normwise relative backward error which is bounded by d^{-q}.

We have already seen that the estimate of Theorem 3.8 is tight, but which matrices exhibit the slowest possible decay rate? We can answer this question for tridiagonal matrices.

THEOREM 3.9. *Let $\{(a_i, b_i, c_i)\}_{i=1}^n$ be a finite sequence, such that $a_i \neq 0$, and*

$$\max_{i=1,\ldots,n} \frac{|b_i| + |c_i|}{|a_i|} = \epsilon < 1.$$

If the vector $x = (x_1, x_2, \ldots, x_n)^T$ given by

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & & \\ c_2 & \ddots & \ddots & \\ & \ddots & & b_{n-1} \\ & & c_n & a_n \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_n \end{bmatrix}$$

exhibits the smallest possible decay rate, i.e., if $|x_1| = \epsilon^n, \epsilon = d^{-1}$, *then* $c_i = 0$, *for* $i = 1, 2, 3, \ldots, n$ *and* $|b_i| = \epsilon |a_i|$ *for* $i = 1, 2, \ldots n$.

Proof. Mikkelsen [9] gives a direct proof. □

4. Error analysis. In this section we do an error analysis of the truncated SPIKE algorithm. We begin by deriving a few results on Gaussian elimination for systems which are strictly diagonally dominant by rows with degree $d > 1$.

We assume that the original problem has been scaled by rows such that $a_{ii} = 1$. Such a scaling preserves the degree of diagonal dominance, and allows us to estimate

$$\|A_i\|_\infty \leq 1 + d^{-1}, \quad \|A_i^{-1}\|_\infty \leq \frac{1}{1 - d^{-1}}, \quad \text{and} \quad \kappa_\infty(A_i) \leq \frac{d+1}{d-1}.$$

Let u denote the unit roundoff error on the machine, and following Higham [6], we define

$$\gamma_j = \frac{ju}{1 - ju}, \tag{4.1}$$

when $ju < 1$. If A is any matrix, then $B = |A|$ is the matrix given by $b_{ij} = |a_{ij}|$. If A, B are matrices of the same dimension, then we write $A \leq B$, if $a_{ij} \leq b_{ij}$ for all i and j.

If A is a banded matrix with upper and lower bandwidth k, which is diagonally dominant by rows, and if Gaussian elimination runs to completion, then the computed solution $\hat{x}$ to $Ax = f$ satisfies

$$(A + \Delta A)\hat{x} = f, \quad |\Delta A| \leq \gamma_{3k+2} |\hat{L}||\hat{U}|,$$

where $\hat{L}$ and $\hat{U}$ are the computed LU factors.

Now, how large is $\|\Delta A\|_\infty$? If A is any n by n matrix and if $A = LU$ is the exact LU factorization, then

$$|L||U| = |AU^{-1}||U| \leq |A||U^{-1}||U|.$$

If U is diagonally dominant by rows, then by Lemma 8.8 [6]

$$\| |U^{-1}||U| \|_\infty \leq (2n - 1). \tag{4.2}$$

This estimate is tight. However, if A is strictly diagonally dominant by rows with degree $d > 1$, then we may be able to improve upon it. By Theorem 3.1 U is strictly diagonally dominant by rows with degree no less than d. Write $U = DV$, where D is the main diagonal of U, then

$$|U^{-1}||U| = |V^{-1}D^{-1}||DV| = |V^{-1}||V|,$$

which allows us to estimate

$$\| |U^{-1}||U| \|_\infty \leq \frac{d+1}{d-1}, \tag{4.3}$$

because $\|I - V\|_\infty \leq d^{-1} < 1$.

It is important to realize that neither (4.2) nor (4.3) need apply to the computed LU factorization, because, while $\hat{L}\hat{U}$ is the exact LU factorization of the matrix $A + \Delta A$, this matrix need not be diagonally dominant! However, since $\hat{L} \to L$, and $\hat{U} \to U$

as $u \to 0$, then $A + \Delta A$ will be strictly diagonally dominant by rows with degree close to d, for u sufficiently small, and then we may estimate

$$\|\Delta A\|_\infty \leq \gamma_{3k+2} \||\hat{L}||\hat{U}|\|_\infty \lesssim \gamma_{3k+2} \frac{d+1}{d-1} \|A\|_\infty.$$

In the following we assume that we may estimate

$$\|\Delta A\|_\infty \leq \gamma_{3k+2} \frac{d+1}{d-1} \|A\|_\infty.$$

Now, what can be said about the solution $\hat{X}$ to the equation $AX = F$ where X and F have m columns? We have

$$(A + \Delta A_j)\hat{x}_j = f_j, \quad |\Delta A_j| \leq \gamma_{3k+2} |\hat{L}||\hat{U}|, \quad j = 1, 2, \dots m,$$

where the perturbations ΔA_j depend on j, but share a common bound which is independent of j. Now, if the unit roundoff error is sufficiently small, specifically if

$$\alpha = \gamma_{3k+2} \left(\frac{d+1}{d-1} \right)^2 < 1, \tag{4.4}$$

then $I + A^{-1}\Delta A_j$ and $I + \Delta A_j A^{-1}$ are both invertible and we may write

$$\hat{x}_j = \sum_{i=0}^{\infty} \left(-A^{-1}\Delta A_j \right)^i x_j = A^{-1} \sum_{i=0}^{\infty} \left(-(\Delta A_j)A^{-1} \right)^i f_j,$$

from which it follows immediately that

$$|\hat{x}_j - x_j| \leq E_1 |x_j|, \quad E_1 = \sum_{i=1}^{\infty} \left(\gamma_{3k+2} |A^{-1}||\hat{L}||\hat{U}| \right)^i,$$
$$|A\hat{x}_j - f_j| \leq E_2 |f_j|, \quad E_2 = \sum_{i=1}^{\infty} \left(\gamma_{3k+2} |\hat{L}||\hat{U}||A^{-1}| \right)^i,$$

which implies

$$|\hat{X} - X| \leq E_1 |X|, \quad \text{and} \quad |A\hat{X} - F| \leq E_2 |F|.$$

The two operators, E_1 and E_2, share a common bound, namely,

$$\|E_1\|_\infty \leq \frac{\alpha}{1-\alpha}, \quad \text{and} \quad \|E_2\|_\infty \leq \frac{\alpha}{1-\alpha},$$

where α is defined by (4.4). It follows that

$$\|\hat{X} - X\|_\infty \leq \frac{\alpha}{1-\alpha} \|X\|_\infty, \quad \text{and} \quad \|A\hat{X} - F\|_\infty \leq \frac{\alpha}{1-\alpha} \|F\|_\infty. \tag{4.5}$$

Stage 1 Each matrix A_i has dimension μ and is strictly diagonally dominant by rows. The computed LU factorization satisfies

$$A_i + \Delta A_i = \hat{L}_i \hat{U}_i, \quad |\Delta A_i| \leq \gamma_{k+1} \left| \hat{L}_i \right| \left| \hat{U}_i \right|,$$

where

$$\left\|\left|\hat{L}_i\right|\left|\hat{U}_i\right|\right\|_\infty \lesssim \frac{d+1}{d-1}\|A_i\|_\infty,$$

when the unit roundoff error u is sufficiently small. We have the same type of estimate for the computed UL factorizations.

Stage 2 In the truncated SPIKE algorithm, we do not compute the entire SPIKE matrix but stop substituting as soon as the truncated reduced system matrix has been computed. However, in order to estimate the error, it is convenient to consider the computation of the entire SPIKE matrix S.

By applying (4.5) repeatedly to the individual block rows we find

$$\|\hat{S}-S\|_\infty \le \frac{2\alpha}{1-\alpha}\|S-I\|_\infty, \quad \left\|D\hat{S}-A\right\|_\infty \le \frac{2\alpha}{1-\alpha}\|A-D\|_\infty.$$

The extra factor of 2 is introduced because we have to treat the superdiagonal and the subdiagonal spikes separately.

Similarly we find for the computation of the modified right-hand side that

$$\|\hat{g}-g\|_\infty \le \frac{\alpha}{1-\alpha}\|g\|_\infty, \quad \text{and} \quad \left\|D\hat{g}-f\right\|_\infty \le \frac{\alpha}{1-\alpha}\|f\|_\infty.$$

It is clear that since $\hat{T}-T$ is a submatrix of $\hat{S}-S$ we have

$$\left\|\hat{T}-T\right\|_\infty \le \left\|\hat{S}-S\right\|_\infty \le \frac{2\alpha}{1-\alpha}\|S-I\|_\infty \le \frac{2\alpha}{1-\alpha}d^{-1}.$$

Stage 3 By Theorem 3.8 the truncated reduced system is a good approximation of the reduced system if d is not too close to 1 and if the partitions are not too small. By Theorem 3.3 the truncated reduced system is strictly diagonally dominant by rows with a degree no less than the original system. It consists of $p-1$ independent systems which are of dimension $2k$. By Theorem 9.3 [6] it follows that if Gaussian elimination runs to completion, then the computed solution $\hat{x}_{tr}$ of the computed truncated reduced system $\hat{T}x_{tr}=\hat{g}_r$ satisfies

$$\left(\hat{T}+\Delta\hat{T}\right)\hat{x}_{tr}=\hat{g}_r, \quad \left|\Delta\hat{T}\right| \le \gamma_{6k}\left|\hat{L}_t\right|\left|\hat{U}_t\right|,$$

where $\hat{L}_t\hat{U}_t$ is the computed LU factorization of the computed truncated reduced system matrix $\hat{T}$. It follows that

$$\|\hat{x}_{tr}-x_{tr}\|_\infty \le \frac{\beta}{1-\beta}\|x_{tr}\|_\infty \quad \text{and} \quad \left\|\hat{T}\hat{x}_{tr}-\hat{g}_r\right\| \le \frac{\beta}{1-\beta}\|\hat{g}_r\|_\infty,$$

provided the unit round off error is so small that

$$\beta = \gamma_{6k}\left(\frac{d+1}{d-1}\right)^2 < 1.$$

Stage 4 Adjusting the original right-hand side, i.e., computing

$$h_i = f_i - C_i x_{i-1}^{(b)} - B_i x_{i+1}^{(t)},$$

introduces a small forward error. Notice that C_i affects only the top of f_i and B_i affects only the bottom of f_i. The componentwise relative forward error is no more than

$$|\hat{h}_i - h_i| \le \gamma_{k+1}\left(|f_i| + |C_i||x_{i-1}^{(b)}| + |B_i||x_{i+1}^{(t)}|\right),$$

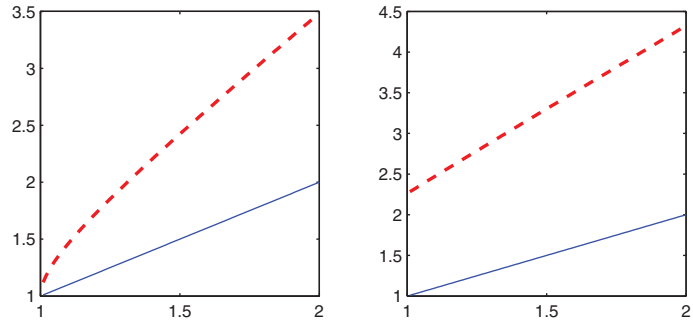


FIG. 5.1. *The degree of diagonal dominance for the matrix $S^{(k)}$ as a function of the degree of diagonal dominance of the original matrices: $A^{(k)}$ (left), and $B^{(k)}$ (right). The matrices are defined by equation* (5.1). *The red dotted line is the experimental result and the solid blue line is the theoretical lower bound.*

regardless of the order in which the scalar products are evaluated. This is an overestimate which does not take into account that the central components of f_i are not changed at all. The solution of the final set of linear equations is identical to stage 2 and generates a normwise relative residual of at most $\frac{\alpha}{1-\alpha}$, as well as a normwise relative forward error of at most $\frac{\alpha}{1-\alpha}$; cf. (4.5).

In short, if d is not too close to 1 and if the partitions are not too small, then the errors at every stage of the algorithm are small. We found that the simplest way to evaluate the overall error was to calculate the residual and estimate

$$\|x-y\|_\infty \leq \|A^{-1}\|_\infty \|f-Ay\|_\infty \leq \frac{1}{1-d^{-1}}\|f-Ay\|_\infty,$$

which turned out to be fairly effective as long as d is not too close to 1.

5. Numerical experiments. We ran experiments to verify the main results of this paper as well as compare the accuracy of the truncated SPIKE algorithm with the algorithm implemented in ScaLAPACK.

5.1. The matrices S, R, and T. We wanted to verify that the degree of diagonal dominance of the SPIKE matrix S was no less than that of the original matrix A. We selected two sequences of matrices with $(n, k_u, k_l) = (10^6, 5, 5)$:

$$A_{ij}^{(k)} = \begin{cases} 1+0.01k & \text{for } i=j \\ -0.1 & \text{for } 0<|i-j|\leq 5, \\ 0 & \text{otherwise} \end{cases} \qquad B_{ij}^{(k)} = \begin{cases} 1+0.01k & \text{for } i=j \\ 0.1 & \text{for } 0<|i-j|\leq 5, \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

for $k = 1, 2, \ldots, 100$. We selected $p = 8$ partitions and a uniform block size of $1.25 \cdot 10^5$. We explicitly computed the entire SPIKE matrix S and the excess $\|S-I\|_\infty$ for each of these 200 matrices, from which we determined the degree of diagonal dominance as $d(S) = 1/\|S-I\|_\infty$. Our results are displayed in Figure 5.1. We found that not only is the degree of diagonal dominance preserved, i.e., $d(S) \geq d(A)$, but there can be a substantial increase in diagonal dominance as well.

We extracted the truncated reduced system matrix T from each of the 200 matrices and computed the condition number in the infinity norm by explicitly inverting T and calculating $\|T^{-1}\|_\infty$. We then plotted the condition number of T as a function of the degree of diagonal dominance of A. The results are displayed in Figure 5.2.

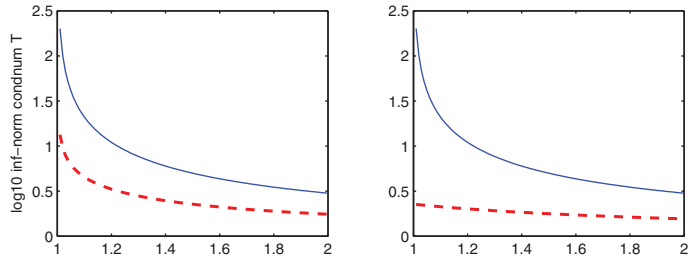


FIG. 5.2. *The condition number of the truncated reduced system as a function of the degree of diagonal dominance of the original system matrices: $A^{(k)}$ (left), and $B^{(k)}$ (right). The matrices are defined by equation* (5.1). *The dotted red line is the experimental result and the solid blue line is the theoretical upper bound.*

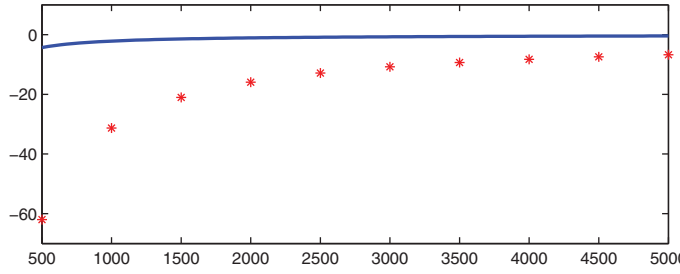


FIG. 5.3. *The infinity norm of the truncation error as a function of the number of partitions. Solid blue line is the theoretical upper bound, while red dots are experimental results. The matrix has $d = 1.01$ and is tridiagonal.*

The theoretical upper bound is given by $\frac{d+1}{d-1}$ where $d = d(A)$ is the degree of diagonal dominance of A. We found that the truncated reduced system was even better conditioned than expected.

We wanted to investigate the size of the truncation error as a function of the degree of diagonal dominance of the original matrix A and the number of partitions p. We selected a tridiagonal Toeplitz matrix with $n = 5 \cdot 10^5$ and 1.01 on the main diagonal and 0.5 on the off-diagonal elements. We choose $p = 500j$, for $j = 1, 2, \ldots, 10$ and computed the truncation error explicitly. The theoretical upper bound is given by d^{-q} where $d = 1.01$ and $q = \lfloor 5 \cdot 10^5/p \rfloor$. The results are displayed in Figure 5.3. The truncation error is much smaller than the theoretical upper bound and it is smaller than the unit roundoff error $u = 2^{-53} \approx 1.1 \cdot 10^{-16}$ as long as $p \leq 2000$.

5.2. The error analysis. We wanted to verify the bounds presented in section 4. We constructed matrices which were diagonally dominant by rows with different degrees and ran them through our implementation of the truncated SPIKE algorithm. The matrices all had $(n, k_l, k_u) = (10^6, 10, 10)$ with every diagonal entry equal to 1. The nonzero, off-diagonal entries where positive and constant for each matrix, such that the degree of diagonal dominance varied from 1.1 for the first matrix to 2.0 for the last matrix, with steps of 0.1. The right-hand side side was generated from the solution which was selected as $x = (1, 1, \ldots, 1)^T$. Our results are listed as Table 5.1 and Table 5.2. The bounds were computed as follows:

1. The modified right-hand side,

$$\|D\hat{g} - f\|_\infty \leq \frac{\alpha}{1-\alpha}\|f\|_\infty.$$

TABLE 5.1
A comparison of certain measurable quantities and their bounds for 10 *different matrices distinguished by their degree of diagonal dominance.*

d	α	$\|D\hat{g} - f\|_\infty$		$\left\|D\hat{S} - A\right\|_\infty$	
		measured	bound	measured	bound
1.1	1.57e − 12	9.58e − 16	2.99e − 12	6.94e − 17	2.85e − 12
1.2	4.30e − 13	1.25e − 15	7.88e − 13	5.90e − 17	7.16e − 13
1.3	2.09e − 13	1.57e − 15	3.69e − 13	7.05e − 17	3.21e − 13
1.4	1.28e − 13	1.51e − 15	2.19e − 13	6.94e − 17	1.83e − 13
1.5	8.88e − 14	1.64e − 15	1.48e − 13	5.11e − 17	1.18e − 13
1.6	6.67e − 14	9.78e − 16	1.08e − 13	5.55e − 17	8.34e − 14
1.7	5.29e − 14	1.51e − 15	8.39e − 14	2.93e − 17	6.22e − 14
1.8	4.35e − 14	1.47e − 15	6.77e − 14	4.47e − 17	4.84e − 14
1.9	3.69e − 14	1.75e − 15	5.63e − 14	4.27e − 17	3.88e − 14
2.0	3.20e − 14	1.30e − 15	4.80e − 14	4.16e − 17	3.20e − 14

TABLE 5.2
A comparison of certain measurable quantities and their bounds for 10 *different matrices distinguished by their degree of diagonal dominance.*

d	$\left\|\hat{T}\hat{x}_{tr} - g_r\right\|_\infty$		$\|A\hat{x} - f\|_\infty$	$\|\hat{x} - x\|_\infty$	
	measured	bound	measured	measured	bound
1.1	7.77e − 16	1.40e − 13	8.88e − 16	8.88e − 16	9.77e − 15
1.2	7.77e − 16	7.33e − 14	1.33e − 15	1.11e − 15	7.99e − 15
1.3	5.55e − 16	5.11e − 14	1.55e − 15	1.55e − 15	5.44e − 15
1.5	4.44e − 16	3.33e − 14	1.55e − 15	1.55e − 15	4.66e − 15
1.6	7.77e − 16	2.89e − 14	8.88e − 16	8.88e − 16	2.37e − 15
1.7	5.55e − 16	2.57e − 14	1.55e − 15	1.22e − 15	3.77e − 15
1.8	5.55e − 16	2.33e − 14	1.55e − 15	1.55e − 15	3.50e − 15
1.9	6.66e − 16	2.15e − 14	1.78e − 15	1.55e − 15	3.75e − 15
2.0	5.55e − 16	2.00e − 14	1.33e − 15	1.33e − 15	2.66e − 15

2. The SPIKE matrix,

$$\left\|D\hat{S} - A\right\|_\infty \leq 2\frac{\alpha}{1-\alpha}d^{-1}.$$

3. The computed truncated reduced system,

$$\left\|\hat{T}\hat{x}_{tr} - \hat{g}_r\right\|_\infty \leq \gamma_{6k}\frac{d+1}{d-1}\|\hat{x}_{tr}\|_\infty.$$

4. The overall error,

$$\|\hat{x} - x\|_\infty \leq \frac{1}{1-d^{-1}}\|A\hat{x} - f\|_\infty.$$

We see that the modified right-hand side g is computed with a small residual and that the bound becomes increasingly accurate as d becomes larger. The SPIKE matrix is computed with a very small residual and the bound is between 10^3 and 10^5 times too large. The computed reduced system is solved with a very small residual and the bound is between 10^2 and 10^3 times larger. Finally we see that using the residual to estimate the error is very reliable, leading to estimates that are accurate within one order of magnitude.

5.3. Comparisons with ScaLAPACK. We began by comparing the errors in the truncated SPIKE algorithm to ScaLAPACK (PDDBTRF/PDDBTRS) for four

TABLE 5.3

The 2-norm of the absolute error for ScaLAPACK (Sca) (PDDBTRF/PDDBTRS) and the truncated SPIKE (T.S) algorithm for four different banded matrices and different numbers of partitions. The results from LAPACK (DGBTRF/DGBTRS) are listed at the bottom of the table.

	(n, k_l, k_u)							
	$(2e4, 10, 10)$		$(1e5, 10, 10)$		$(1e5, 50, 50)$		$(1e6, 10, 10)$	
p	Sca	T.S	Sca	T.S	Sca	T.S	Sca	T.S
2	$4.98e-10$	$5.02e-10$	$5.33e-9$	$5.34e-9$	$1.33e-8$	$1.33e-8$	$2.10e-7$	$2.10e-7$
4	$4.98e-10$	$5.02e-10$	$5.33e-9$	$5.33e-9$	$1.33e-8$	$1.33e-8$	$2.10e-7$	$2.10e-7$
8	$4.97e-10$	$5.02e-10$	$5.32e-9$	$5.33e-9$	$1.33e-8$	$1.33e-8$	$2.10e-7$	$2.10e-7$
12	$4.97e-10$	$5.01e-10$	$5.32e-9$	$5.33e-9$	$1.33e-8$	$1.34e-8$	$2.10e-7$	$2.10e-7$
16	$4.97e-10$	$5.02e-10$	$5.32e-9$	$5.33e-9$	$1.33e-8$	$1.33e-8$	$2.10e-7$	$2.10e-7$
24	$4.95e-10$	$5.00e-10$	$5.32e-9$	$5.33e-9$	$1.33e-8$	$1.34e-8$	$2.10e-7$	$2.10e-7$
32	$4.95e-10$	$5.02e-10$	$5.32e-9$	$5.33e-9$	$1.32e-8$	$1.33e-8$	$2.10e-7$	$2.10e-7$
48	$4.90e-10$	$4.98e-10$	$5.31e-9$	$5.32e-9$	$1.33e-8$	$1.33e-8$	$2.10e-7$	$2.10e-7$
64	$4.88e-10$	$4.95e-10$	$5.30e-9$	$5.32e-9$	$1.32e-8$	$1.33e-8$	$2.10e-7$	$2.10e-7$
128	$4.81e-10$	$4.88e-10$	$5.28e-9$	$5.30e-9$	$1.33e-8$	$1.34e-8$	$2.10e-7$	$2.10e-7$
256	N/A	$1.43e-7$	$5.23e-9$	$5.26e-9$	$1.33e-8$	$7.64e-2$	$2.10e-7$	$2.10e-7$
LA	$4.99e-10$		$5.33e-9$		$1.33e-8$		$2.10e-7$	

different matrices with

$$(n, k_l, k_u) \in \left\{ \left(2.0 \cdot 10^4, 10, 10\right), \left(10^5, 10, 10\right), \left(10^5, 50, 50\right), \left(10^6, 10, 10\right) \right\}.$$

Every diagonal entry was 1 and all other entries within the band were 10^{-2}. The right-hand side was constructed from the solution which was selected as $(1, 2, \ldots, n)^T$. The number of partitions were 2, 4, 8, 16, 24, 32, 48, 64, 128, and 256. The calculations were carried out in IEEE double precision arithmetic. We measured the 2-norm of the absolute error. Our results are displayed in Table 5.3. In our experiments ScaLAPACK did slightly better than the truncated SPIKE algorithm, but the difference between the two algorithms decreased, as the problems became larger. We would like to draw attention to the case of $p = 256$. In this case ScaLAPACK cannot be applied to the first matrix where $n = 20{,}000$, because the matrix is too small and the bandwidth is large compared to the number of partitions, and the routine issues the appropriate error message. The truncated SPIKE algorithm had a large error for the first and the third matrix. This is due to the fact that the infinity norm of the truncation error was very large: for the first matrix it was $1.62 \cdot 10^{-12}$, while for the third matrix it was $1.52 \cdot 10^{-7}$. In all other cases we found that the infinity norm of the truncation error was either less than machine ϵ or much smaller than the unit round off error u. The experiments with $p = 256$ emphasize the fact that the truncated SPIKE algorithm should not be applied to problems where the partitions are either too small or where the diagonal blocks are not diagonally dominant enough. The first matrix is diagonally dominant with degree $d = 5$, and for $p = 256$ the dimension of the smallest partition was 78. In this case Theorem 3.7 gives an upper bound for the infinity norm of the truncations error of $5^{-7} \approx 1.28 \cdot 10^{-5}$. In other words, we knew in advance that the result might not be accurate. Theorem 3.7 does not apply to the third matrix, which is not strictly diagonally dominant.

We found nine matrices that were diagonally dominant at Matrix Market. They were all quite small, with dimensions no larger than 5000. We extracted narrow banded matrices from these matrices by choosing $k = \lceil 0.01n \rceil$. We ran the examples through LAPACK (DGBTRF/DGBTRS), ScaLAPACK (PDDBTRF/PDDBTRS), our own implementation of the truncated SPIKE algorithm, as well as the SPIKE package itself (TU0). The matrices were scaled such that the main diagonals were 1

TABLE 5.4

The 2-norm of the absolute error for nine different matrices from Matrix Market. The results are given for LAPACK (dgbtrf/dgbtrs) and ScaLAPACK (PDDBTRF/PDDBTRS). The results are given for 2, 4, and 8 partitions.

matrix	n	LA	ScaLAPACK		
			2	4	8
dwb512	512	3.27e − 15	3.14e − 15	3.14e − 15	3.14e − 15
gr_30_30	900	0.00e + 00	0.00e + 00	1.57e − 16	2.94e − 16
jpwh_991	991	1.37e − 15	2.04e − 15	2.03e − 15	2.01e − 15
nos6	675	0.00e + 00	3.05e − 15	3.07e − 15	3.10e − 15
orsirr_1	1030	4.40e − 15	4.44e − 15	4.42e − 15	4.36e − 15
orsirr_2	886	4.11e − 15	4.10e − 15	4.11e − 15	4.12e − 15
orsreg_1	2205	7.08e − 15	7.12e − 15	7.30e − 15	6.93e − 15
sherman3	5005	1.92e − 12	1.99e − 12	1.99e − 12	1.99e − 12
sherman4	1104	2.53e − 15	2.59e − 15	2.58e − 15	2.58e − 15

TABLE 5.5

The 2-norm of the absolute error for nine different matrices from Matrix Market. The results are given for our implementation (T.S) of the truncated SPIKE algorithm, as well as the current implementation of the SPIKE package (TU0). The results are given for 2, 4, and 8 partitions.

matrix	T.S			T.U		
	2	4	8	2	4	8
dwb512	3.30e − 15	3.30e − 15	3.30e − 15	3.04e − 15	3.09e − 15	3.11e − 15
gr_30_30	0.00e + 00	0.00e + 00	0.00e + 00	0.00e + 00	1.11e − 16	5.09e − 16
jpwh_991	2.00e − 15	1.97e − 15	1.95e − 15	2.03e − 15	2.06e − 15	2.01e − 15
nos6	0.00e + 00	0.00e + 00	0.00e + 00	3.03e − 15	3.12e − 15	3.01e − 15
orsirr_1	4.39e − 15	4.38e − 15	4.31e − 15	4.40e − 15	4.26e − 15	4.15e − 15
orsirr_2	4.07e − 15	4.10e − 15	4.10e − 15	4.16e − 15	3.94e − 15	4.06e − 15
orsreg_1	7.16e − 15	7.16e − 15	7.00e − 15	6.49e − 15	6.79e − 15	6.18e − 15
sherman3	1.99e − 12	1.99e − 12	1.99e − 12	1.98e − 12	1.98e − 12	1.98e − 12
sherman4	2.49e − 15	2.50e − 15	2.51e − 15	2.44e − 15	2.41e − 15	2.48e − 15

and the right-hand side was generated from the solution which was selected as $x = (1, 1, \ldots, 1)^T$. We measured the 2-norm of the absolute error. Our results are listed in Table 5.4 and Table 5.5. We found no substantial difference in the accuracy of the four different routines.

6. Conclusion. We have shown that the SPIKE matrix is diagonally dominant by rows with a degree no less than that of the original matrix. We have derived a tight upper bound on the truncation error for the general case. We showed that the error committed at each stage is small, and we found that our bounds are probably pessimistic. We compared the truncated SPIKE algorithm to the corresponding algorithm in ScaLAPACK (PDDBTRF/PDDBTRS) and found no substantial difference between the accuracy of the two methods. The advantage of the truncated SPIKE algorithm is that if the matrix is diagonally dominant by rows with degree $d > 1$ and the partitions are sufficiently large, then the reduced system is essentially block diagonal and can be solved with a constant amount of communication, with all but one processor contributing equally to the solution of the reduced system. In ScaLAPACK (PDDTRS) the reduced system is solved recursively with the number of active processors being cut in half at each iteration.

Acknowledgments. The authors wish to thank the referees as well as the editor for their comments and suggestions which improved the presentation. The authors also want to thank their advisor Ahmed Sameh for his continued support.

REFERENCES

[1] P. Arbenz, A. Cleary, J. Dongarra, and M. Hegland, *A comparison of parallel solvers for diagonally dominant and general narrow banded linear systems* II, in EuroPar '99 Parallel Processing, P. Amestoy, Ph. Berger, M. Dayd, I. Duff, V. Frayss, L. Giraud, D. Ruiz, eds., Springer, Berlin, 1999, pp. 1078–1087.

[2] S.C. Chen, D.J. Kuck, and A. Sameh, *Practical parallel band triangular system solvers*, ACM Trans. Math. Software, 4 (1978), pp. 270–277.

[3] S. Demko, W.F. Moss, and Ph.W. Smith, *Decay rates for inverses of band matrices*, Math. Comput., 43 (1984), pp. 491–499.

[4] J.J. Dongarra and A. Sameh, *On some parallel banded system solvers*, Parallel Comput., 1 (1984), pp. 223–235.

[5] A. George and Kh. Ikramov, *Gaussian elimination is stable for the inverse of a diagonally dominant matrix*, Math. Comp., 73 (2003), pp. 653–657.

[6] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, PA, 2002.

[7] J.-L. Larriba-Pey, À. Jorba, and J.J. Navarro, *Spike algorithm with savings for strictly diagonal dominant tridiagonal systems*, Microprocessing and Microprogramming, 39 (1993), pp. 125–128.

[8] D.H. Lawrie and A. Sameh, *The computation and communication complexity of a parallel banded system solver*, ACM Trans. Math. Software, 10 (1984), pp. 185–195.

[9] C.C.K. Mikkelsen, *The Decay Rate of the Solution to a Tridiagonal Linear System with a Very Special Right Hand Side*, Technical report, CSD TR #08-021, Computer Science Department, Purdue University, West Lafayette, IN, 2008.

[10] E. Polizzi and A. Sameh, *A parallel hybrid banded system solver: The SPIKE algorithm*, Parallel Comput., 32 (2006), pp. 177–194.

[11] E. Polizzi and A. Sameh, *SPIKE: A parallel enviroment for solving banded linear systems*, Comput. Fluids, 36 (2007), pp. 113–120.

[12] A.H. Sameh and D.J. Kuck, *On stable parallel linear system solvers*, J. ACM, 25 (1978), pp. 81–91.

[13] X.-H. Sun, H. Zhang, and L.M. Ni, *Efficient tridiagonal solvers on multicomputers*, IEEE Trans. Comput., 41 (1992), pp. 286–296.

[14] X.-H. Sun, *Application and accuracy of the parallel diagonal dominant algorithm*, Parallel Comput., 21 (1995), pp. 1241–1267.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1520–1545

SYMBOLIC AND EXACT STRUCTURE PREDICTION FOR SPARSE GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING*

LAURA GRIGORI†, JOHN R. GILBERT‡, AND MICHEL COSNARD§

Abstract. In this paper we consider two structure prediction problems of interest in Gaussian elimination with partial pivoting of sparse matrices. First, we consider the problem of determining the nonzero structure of the factors L and U during the factorization. We present an exact prediction of the structure that identifies some numeric cancellations appearing during Gaussian elimination. The numeric cancellations are related to submatrices of the input matrix A that are structurally singular, that is, singular due to the arrangements of their nonzeros, and independent of their numerical values. Second, we consider the problem of estimating upper bounds for the structure of L and U prior to the numerical factorization. We present tight exact bounds for the nonzero structure of L and U of Gaussian elimination with partial pivoting $PA = LU$ under the assumption that the matrix A satisfies a combinatorial property, namely, the Hall property, and that the nonzero values in A are algebraically independent of each other. This complements existing work showing that a structure called the row merge graph represents a tight bound for the nonzero structure of L and U under a stronger combinatorial assumption, namely, the strong Hall property. We also show that the row merge graph represents a tight symbolic bound for matrices satisfying only the Hall property.

Key words. sparse LU factorization, partial pivoting, structure prediction, characterization of fill

AMS subject classifications. 65F50, 65F05, 68R10

DOI. 10.1137/050629343

1. Introduction.

In this paper we consider the problem of structure prediction when solving a linear system $Ax = b$ by Gaussian elimination with partial pivoting, where A is an $n \times n$ sparse, nonsingular, and nonsymmetric matrix and b is an n-vector. This elimination, also called LU factorization, involves explicit factorization of the matrix A into the product of L and U, where L is a unit lower triangular matrix and U is an upper triangular matrix.

One of the main characteristics of the sparse LU factorization is the notion of fill. "Fill" denotes a nonzero entry in the factors that was a zero in matrix A. When Gaussian elimination without pivoting is used, the nonzero structure of the factors can be computed without referring to the numerical values of the matrix and is determined before performing the numerical computation of the factors themselves. Knowledge of this structure is used to allocate memory, set up data structures, schedule parallel tasks, and save time [16] by avoiding operations on zeros. When pivoting is used for numerical stability [13], the structure of L and U depends not only on the structure of A but also on the row interchanges. As the row interchanges are determined while doing the numerical factorization, the computation of the structure of the factors has to be interleaved with the computation of the numerical values of the factors. Prior to the numerical factorization, only upper bounds of the structure of L and U can be determined.

*Received by the editors April 18, 2005; accepted for publication (in revised form) by E. Ng July 17, 2008; published electronically December 19, 2008.
http://www.siam.org/journals/simax/30-4/62934.html

†INRIA Saclay - Ile-de-France, Laboratoire de Recherche en Informatique, Paris-Sud 11 University, Bat 490, 91405 Orsay, France (Laura.Grigori@inria.fr).

‡Computer Science Department, University of California at Santa Barbara, Santa Barbara, CA 93106-5110 (gilbert@cs.ucsb.edu).

§INRIA Sophia Antipolis, 2004 route des Lucioles BP93, 06902 Sophia Antipolis, France (Michel.Cosnard@inria.fr).

We discuss in this paper two structure prediction problems. The first problem considers the computation of the nonzero structure of the factors during Gaussian elimination with row interchanges. The second problem is to obtain tight bounds of the structure of L and U prior to the numerical factorization. For both problems, we study relations between the combinatorial properties of the nonzero structure of the matrix A and the LU factorization.

Two kinds of structure prediction and two combinatorial properties of the input matrix are usually considered for these problems. The two structure predictions are called *symbolic* and *exact* [11]. *Symbolic* structure prediction assumes that the addition or subtraction of two nonzero results always yields a nonzero result. It ignores possible numeric cancellations occurring during the LU factorization. *Exact* structure prediction assumes that the nonzero values in A are algebraically independent from each other; in other words, it assumes that any computed zero is due to combinatorial properties of the nonzero structure. The two combinatorial properties of the input matrix are called the *strong Hall* property and the *Hall* property. The *strong Hall* property is an irreducibility condition. The *Hall* property is a weaker combinatorial assumption and is related to matrices with full-column rank. We will define these two properties in more detail later in the paper. A matrix that satisfies the Hall property can be decomposed using the Dulmage–Mendelsohn decomposition [2, 17, 18] into a block upper triangular form such that every block on the diagonal satisfies the strong Hall property. However, in practice this decomposition is not always used, and hence it is interesting to understand the structure prediction for matrices satisfying either the strong Hall property or the Hall property.

Much of the research has been aimed at predicting the structure and bounds of the factors L and U as tightly as possible [9, 10, 11, 12, 20]. The existing results for determining the nonzero structure of L and U during Gaussian elimination with partial pivoting $PA = LU$ are symbolic [20]. Under several additional conditions, this structure prediction is exact [11]. But in general it ignores possible numeric cancellations during the factorization for matrices satisfying the strong Hall property or only the Hall property. For the problem of predicting bounds for the structure of L and U prior to the numerical factorization, the existing results in the literature assume that A satisfies the strong Hall property. The results assume the LU factorization with partial pivoting is seen as

$$A = P_1 L_1 P_2 L_2 \dots P_{n-1} L_{n-1} U,$$

where P_i is an $n \times n$ elementary permutation matrix identifying the row interchanges at step i. L_i is an $n \times n$ elementary lower triangular matrix whose ith column contains the multipliers at step i. U is an $n \times n$ upper triangular matrix. $\widetilde{L}$ is the $n \times n$ matrix whose ith column is the ith column of L_i so that $\widetilde{L} - I = \sum_i (L_i - I)$. Note that this $\widetilde{L}$ is not the same as the factor L obtained from the factorization $PA = LU$. Both matrices are unit lower triangular, and they contain the same nonzero values but in different positions. The factor L has its rows in the order described by the entire row permutations. The factor $\widetilde{L}$ has the rows of its ith column in the order described by only the first i row interchanges. George and Ng [9] predict an upper bound of the nonzero structure of $\widetilde{L}$ and U, called the row merge graph, that contains the nonzeros in $\widetilde{L}$ and U for all possible row permutations which can later appear in the numerical factorization due to pivoting. Gilbert and Ng [11] showed that this is a tight exact bound for a square matrix with nonzero diagonal which satisfies the strong Hall property.

In this paper we provide answers to several open questions related to the two structure prediction problems considered here. For the first problem, we identify the exact structure prediction of L and U during LU factorization with partial pivoting. For the second problem, we describe the exact bounds of the factors obtained from the factorization $PA = LU$, when the matrix A satisfies only the Hall property. These exact bounds are not symbolic bounds. Then we show that the row merge graph represents symbolic bounds for the structure of $\widetilde{L}$ and U.

The exact structure prediction is based on the following approach: all of the elements of the factors L and U can be computed using the determinants of two submatrices of the input matrix A (see, for example, Gantmacher [8]). Consider, for example, the element in position (i, j) of U, where i and j are two indices with $i \leq j$. Let A_{i-1} be the submatrix of A formed by the first $(i-1)$ columns and the first $(i-1)$ rows of A. Let K be the $i \times i$ submatrix of A that includes the first i rows, the first $i-1$ columns, and column j of A. Then the value in position (i, j) of the factor U is given by the quotient of the determinant of K and the determinant of A_{i-1}. A similar relation exists for the elements of L. Our new results identify when the submatrix K is structurally singular, that is, singular due to the arrangements of its nonzeros, and independent of the numerical values. In exact arithmetic, the determinant of K is zero, and hence the element in position (i, j) corresponds to a numeric cancellation. This numeric cancellation is identified in our new results on exact structure prediction. However, in a backward stable factorization $A + E = \hat{L}\hat{U}$, the computed factors $\hat{L}$ and $\hat{U}$ are not necessarily close to the exact $A = LU$ factors, even though the norm of E is small. In particular, a zero in L or U may, in principle, be large in $\hat{L}$ or $\hat{U}$, so rounding it to zero may cause backward instability.

The rest of the paper is organized as follows. In section 2 we present background and several new results used throughout the paper. In section 3 we consider the problem of determining the nonzero structure of the factors L and U during Gaussian elimination with partial pivoting. We present new results that give an exact characterization of the fill occurring in the LU factorization. We show how the theoretical results can be used in an algorithm for computing fill-ins.

In sections 4 and 5 we consider the problem of predicting bounds for the structure of L and U prior to the numerical factorization. In section 4 we present an exact analysis for matrices that satisfy the Hall property. We present tight exact bounds for the nonzero structure of L and U of Gaussian elimination with partial pivoting $PA = LU$. In section 5 we present a symbolic analysis, and we show that the row merge graph is a lower symbolic bound for the factors $\widetilde{L}$ and U of the factorization $A = P_1 L_1 P_2 L_2 \dots P_{n-1} L_{n-1} U$. In other words, for every edge of the row merge graph of a Hall matrix, there is a permutation such that this edge corresponds to a symbolic nonzero in $\widetilde{L}$ or U. By a simple counterexample, we will show that the row merge graph is not a tight bound for the factors L and U in the exact sense. These results are of practical interest since the row merge graph is used by several solvers implementing the sparse LU factorization with partial pivoting. In solvers like the sequential and shared memory versions of SuperLU [5, 6], the row merge graph is used to estimate the memory needs prior to the LU factorization. In solvers proposed in [9, 21], the numerical computation of the factors L and U is performed on the row merge graph, and some operations involve zero elements. Finally, section 6 presents concluding remarks.

2. Graphs of matrices and their properties. In this section we provide the necessary notions to study the structure prediction of the sparse LU factorization

with partial pivoting. We give definitions, previously published results, and two new results (Lemmas 2.6 and 2.7) that are needed by our subsequent proofs.

Let A be a sparse $n \times n$ matrix. A_{ij} denotes the element at row i and column j of A. We refer to the determinant of matrix A as $\det(A)$. We denote the submatrix of A formed by elements of row indices from i to j and column indices from d to e as $A(i\colon j, d\colon e)$. When the indices are not consecutive, we use the following notation: $A([i\colon j, k], d\colon e)$ denotes the submatrix of A formed by elements of row indices from i to j and k and column indices from d to e. We refer to the submatrix $A(1\colon i, 1\colon i)$ as the principal minor of order i of A.

Two graphs are used to predict the nonzero structure of the factors L and U from the structure of A. The first graph is the directed graph of A and is denoted by $G(A)$. This graph has n vertices and an edge $\langle i, j\rangle$ for each nonzero element A_{ij}. We say that the edge $\langle i, j\rangle$ is incident on the vertices i and j.

The second graph is the bipartite graph of A, denoted by $H(A)$. This graph is undirected and has n row vertices, n column vertices, and an edge $\langle i', j\rangle$ if and only if the element A_{ij} is nonzero. Note that whenever possible, we use prime to distinguish between row vertices and column vertices in a bipartite graph. Also we use i, j, k, d, and e to denote a vertex of H for which it is known if it is a column or a row vertex. That is, i' stands for a row vertex and i for a column vertex. We use v and w to denote a generic vertex of H, that is, a vertex that can be a row vertex or a column vertex.

A *path* is a sequence of distinct vertices $\mathcal{Q} = (v_0, v_1, \ldots, v_{q-1}, v_q)$ such that for each two consecutive vertices v_i, v_{i+1} there is an edge from v_i to v_{i+1}. The length of this path is q. The vertices $v_1, \ldots, v_{q-1}$ are called intermediate vertices.

Let H be a bipartite graph with m row vertices and n column vertices. A matching M on H is a set of edges, no two of which are incident on the same vertex. A vertex is covered or matched by M if it is an end point of an edge of M. A matching is called column-complete if it has n edges, row-complete if it has m edges, and perfect if $m = n$ and it is both row- and column-complete. Given a graph H and a column vertex i, we denote by $H - i$ the subgraph of H induced by all of the row vertices and all of the column vertices except i.

The next lemma identifies a matching in the bipartite graph H of A such that if the edges of M become the diagonal elements, the values chosen make the permuted matrix strongly diagonally dominant. It will be used in section 4 to prove our results on exact structure prediction for Hall matrices.

LEMMA 2.1 (Gilbert and Ng [11]). *Suppose the bipartite graph H has a perfect matching M. Let A be a matrix with $H(A) = H$ such that $A_{ij} > n$ for $\langle i', j\rangle \in M$ and $0 < A_{ij} < 1$ for $\langle i', j\rangle \notin M$. If A is factored by Gaussian elimination with partial pivoting, then the edges of M will be the pivots.*

If M is a matching on H, an alternating path with respect to M is a path on which every second edge is an element of M. A c-alternating path is a path that follows matching edges from rows to columns. An r-alternating path is a path that follows matching edges from columns to rows. Suppose the last vertex of one c-alternating path is the first vertex of another c-alternating path. The path obtained by their concatenation is also a c-alternating path. The same result holds for r-alternating paths. Suppose $\mathcal{Q}$ is an alternating path from an unmatched vertex v to a different vertex w. If the last vertex w on $\mathcal{Q}$ is unmatched or the last edge on $\mathcal{Q}$ belongs to M, then a new matching M_1 can be obtained from M by alternating along path $\mathcal{Q}$. The set of edges of M_1 is given by $M \oplus \mathcal{Q} = (M \cup \mathcal{Q}) - (M \cap \mathcal{Q})$. If w is matched by M, then v is matched and w is unmatched by M_1 and $|M_1| = |M|$. If w is unmatched

by M, then both v and w are matched by M_1, $|M_1| = |M| + 1$, and $\mathcal{Q}$ is called an augmenting path with respect to M.

2.1. Hall and strong Hall graphs. We briefly review the Hall and the strong Hall properties and related results. For a detailed description of Hall and strong Hall matrices and their properties, the reader is directed to [2, 3, 11].

A bipartite graph with m rows and n columns has the *Hall property* if every set of k column vertices is adjacent to at least k row vertices, for all $1 \leq k \leq n$. The next theorem and corollary relate the Hall property to column-complete matchings and matrices with full-column rank. In Corollary 2.3 [11] it is shown that if H is Hall and given a matrix A with $H = H(A)$, then the set of ways to fill in its values to make it singular has measure zero. Hence almost all matrices A with $H = H(A)$ have full-column rank.

THEOREM 2.2 (Hall's theorem). *A bipartite graph has a column-complete matching if and only if it has the Hall property.*

COROLLARY 2.3 (Gilbert and Ng [11]). *If a matrix A has full-column rank, then $H(A)$ is Hall. Conversely, if H is Hall, then almost all matrices A with $H = H(A)$ have full-column rank.*

Known results in structure prediction were obtained under an additional assumption, called the strong Hall property. A bipartite graph with m rows and $n \leq m$ columns satisfies the *strong Hall property* if

(i) $m = n > 1$ and every set of k column vertices is adjacent to *more than* k row vertices, for all $1 \leq k < n$, or

(ii) $m > n$ and every set of k column vertices is adjacent to *more than* k row vertices, for all $1 \leq k \leq n$.

LEMMA 2.4 (Gilbert and Ng [11]). *If H is strong Hall and has more nonzero rows than columns and M is any column-complete matching on H, then from every row or column vertex v of H there is a c-alternating path to some unmatched row vertex i' (which depends on v and M).*

The next theorem relates alternating paths and matchings in strong Hall graphs. This theorem was used in several structure prediction results, in the context of sparse LU factorization by Gilbert and Ng in [11], as well as in the sparsity analysis of QR factorization by Coleman, Edenbrandt, and Gilbert in [4] and Hare et al. in [15]. In this paper we will use it in Lemma 2.6 to derive a new result on alternating paths and matchings in strong Hall graphs.

THEOREM 2.5 (alternating paths, Gilbert [12]). *Let H be a strong Hall graph with at least two rows, let i be a column vertex of H, and let v be any row or column vertex of H such that a path exists from i to v. Then H has a column-complete matching for which there exists a c-alternating path from i to v (or, equivalently, an r-alternating path from v to i).*

The next lemma is new. Given a path in a bipartite graph H between a column vertex and a row vertex or between two row vertices, the lemma shows that there is an alternating path with respect to a column-complete matching of H which excludes a row vertex at the extremity of the path. We will use it in sections 3 and 4 to estimate the nonzero structure of the factors L and U.

LEMMA 2.6. *Let H be a strong Hall graph with more nonzero rows than columns, let v be a row or column vertex of H, and let i' be any row vertex of H such that a path exists from v to i'. Then H has a column-complete matching which excludes vertex i' and for which there exists a c-alternating path from v to i'.*

Proof. We distinguish two different cases.

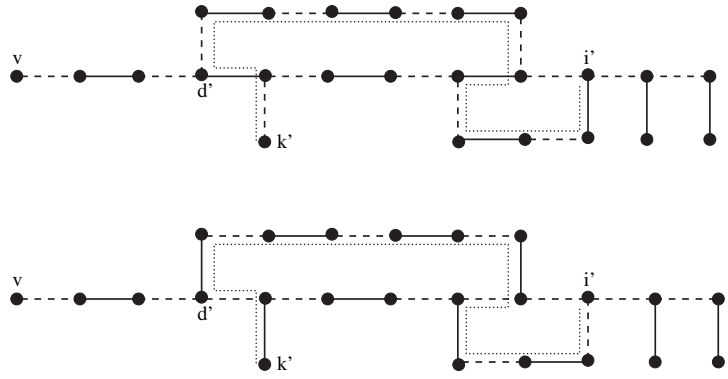


FIG. 2.1. *Case* 1 *of Lemma* 2.6. *In the upper graph, the solid edges are the matching* M*; path* $\mathcal{P}$ *is the horizontal path from* v *to* i'*; path* $\mathcal{Q}$ *is the light dotted line from* i' *to* k'*. In the lower graph, the solid edges are the matching* M_1*. The path obtained by concatenating* $\mathcal{P}[v\colon d']$ *and* $\mathcal{Q}[d'\colon i']$ *is c-alternating with respect to* M_1*.*

Case 1 (v is a column vertex). By hypothesis, there is a path from v to i'. As H is strong Hall, the alternating path Theorem 2.5 applies and says that H has a column-complete matching M for which there exists a c-alternating path $\mathcal{P}$ from v to i'. If i' is not covered by M, then M is the column-complete matching searched. Otherwise, Lemma 2.4 implies that there is an unmatched row vertex k' and a c-alternating path $\mathcal{Q}$ from i' to k'. Now obtain matching M_1 from M by alternating along path $\mathcal{Q}$, where i' is unmatched in M_1.

If $\mathcal{P}$ and $\mathcal{Q}$ have no vertices in common (except row vertex i'), then $\mathcal{P}$ is still c-alternating from v to i' with respect to M_1. If the only vertex in common for $\mathcal{P}$ and $\mathcal{Q}$ (except row vertex i') is column vertex v, then let e' be the row vertex matched by M to v that belongs to the path $\mathcal{Q}$. The path formed by $\langle v, e'\rangle$ followed by $\mathcal{Q}[e'\colon i']$ is c-alternating with respect to M_1.

If $\mathcal{P}$ and $\mathcal{Q}$ have intermediate vertices in common, let d' be the first (row) vertex of $\mathcal{P}$ (starting from v) which belongs to $\mathcal{Q}$. The path obtained by the concatenation of $\mathcal{P}[v\colon d']$ and $\mathcal{Q}[d'\colon i']$ is c-alternating with respect to M_1, and this ends the proof for this case. This case is illustrated in Figure 2.1.

Case 2 (v is a row vertex). We denote the row vertex v as v'. By hypothesis, there is a path from v' to i'. Suppose $v' \neq i'$; otherwise there is nothing to prove. Let d be the first column vertex on this path, that is, the next vertex after v'. H is a strong Hall graph that has a path from column vertex d to row vertex i'. The first case of this theorem, that we have just proved, says that there is a column-complete matching M that excludes vertex i' and for which there exists a c-alternating path $\mathcal{P}$ from d to i'. We distinguish four cases.

Case 2.1 (v' is not matched by M). Let e' be the row vertex matched by M to the column vertex d. We obtain a new matching M_1 by unmatching row vertex e' and matching row vertex v' to row vertex d. The path formed by $\langle v', d\rangle$ followed by $\mathcal{P}$ is c-alternating from v' to i' with respect to M_1. Note that M_1 excludes row vertex i', and this is the path searched.

Case 2.2 (v' is matched by M to the column vertex d). The path obtained by $\langle v', d\rangle$ followed by $\mathcal{P}$ is c-alternating from v' to i' with respect to the matching M, and the matching M excludes row vertex i'.

Case 2.3 (v' is matched by M and belongs to the path $\mathcal{P}$). Then $\mathcal{P}[v'\colon i']$ is a c-alternating path with respect to the matching M.

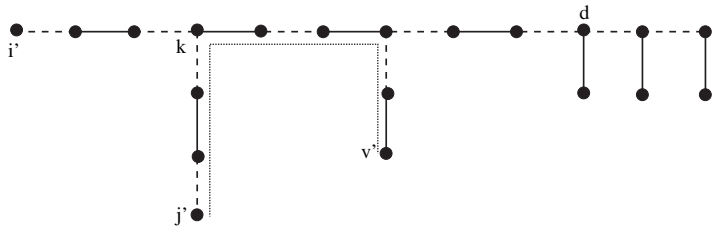


FIG. 2.2. *Case* 2.4 *of Lemma* 2.6. *The solid edges are the matching* M*; path* $\mathcal{P}$ *is the horizontal path from* i' *to* d*; path* $\mathcal{R}$ *is the light dotted line from* v' *to* j'*. Here,* $\mathcal{P}$ *and* $\mathcal{R}$ *have at least one common vertex. Vertex* k *is the last vertex on* $\mathcal{P}$ *(starting from* d*) that belongs to* $\mathcal{R}$*. The path obtained by concatenating* $\mathcal{R}[v':k]$ *and* $\mathcal{P}[k:i']$ *is c-alternating with respect to* M*, and* M *excludes row vertex* i'*.*

Case 2.4 (v' is matched by M to a different column vertex than d and does not belong to the path $\mathcal{P}$). Lemma 2.4 applies and says that there is an unmatched row vertex j' and a c-alternating path $\mathcal{R}$ from v' to j'.

If $\mathcal{P}$ and $\mathcal{R}$ have no vertices in common, then obtain matching M_1 from M by alternating along path $\mathcal{R}$. As v' is matched in M, then v' is unmatched in M_1 and j' is matched in M_1. From here we proceed as in Case 2.1, and we obtain a matching that excludes vertex i' and with respect to which there is a c-alternating path from v' to i'.

If $\mathcal{P}$ and $\mathcal{R}$ have at least one vertex in common, then let k be the last vertex of $\mathcal{P}$ (starting from d) which belongs to $\mathcal{R}$. Note that k has to be a column vertex. The path obtained by concatenating $\mathcal{R}[v':k]$ and $\mathcal{P}[k:i']$ is c-alternating with respect to M, M excludes the row vertex i', and this ends the proof. This case is illustrated in Figure 2.2. □

2.2. Hall sets and their properties. For a bipartite graph H with m row vertices and $n \leq m$ column vertices, a set of k column vertices, $1 \leq k \leq n$, forms a *Hall set* if these columns are adjacent to exactly k rows [15].

Under the assumption that A satisfies the Hall property, the union of two Hall sets is a Hall set, so there exists a unique Hall set of maximum cardinality in any given set of columns. The set of maximum cardinality might be empty. Let C_j be the Hall set of maximum cardinality in the first j columns; we define $C_0 = \emptyset$. Let R_j be the set of all row indices covered by the columns of C_j; thus C_j and R_j have the same cardinality. Note that if we assume all diagonal entries of A are nonzero, then $R_j = \{i' : 1 \leq i \leq j \text{ and } i \in C_j\}$.

The Hall sets of maximum cardinality are useful to partition a Hall graph into two subgraphs: one that satisfies the Hall property and another one that satisfies the strong Hall property. Let H be a bipartite graph with m row vertices and $n < m$ column vertices that satisfies the Hall property. Let C be the Hall set of maximum cardinality in H, and let R be the set of row vertices covered by column vertices of C. The first subgraph $\widetilde{H}$ is induced by all of the row vertices in R and all of the column vertices in C. This subgraph satisfies the Hall property. The second subgraph $\widehat{H}$ is induced by all of the row vertices except those in R and all of the column vertices except those in C. This subgraph is strong Hall because its Hall set of maximum cardinality is empty.

In a similar way, we can partition the edges of a column-complete matching M of H into edges belonging to the graph $\widetilde{H}$ and edges belonging to the graph $\widehat{H}$. This is expressed in a more general way in the following lemma.

LEMMA 2.7. *Let A be an $m \times n$ Hall matrix, $m \geq n$. Let C be a Hall set of cardinality p in A, where $p \leq n$, and let R be the set of all row indices covered by the columns of C. Suppose M is a column-complete matching in the bipartite graph $H(A)$. Then each column vertex j of C is matched by M to a row vertex i' of R.*

Proof. The proof is immediate. □

3. Nonzero structure of L and U during Gaussian elimination with partial pivoting. Let A be an $n \times n$ nonsingular matrix. In this section we consider the problem of determining the nonzero structure of the factors L and U during Gaussian elimination with partial pivoting. In the first part of this section we consider the LU factorization without pivoting. We first present a brief overview of several well-known results described in the literature. Then we describe why these results ignore numeric cancellations related to submatrices of A that are structurally singular. In section 3.1 we present new results that identify some numeric cancellation occurring during Gaussian elimination and caused by submatrices of A that are structurally singular. In section 3.2 we describe how the new results can be used in the Gaussian elimination with partial pivoting. We also present an algorithm that uses the new results to compute the nonzero structure of the factors L and U.

The main result in the structure prediction of Gaussian elimination without pivoting is the fill path Lemma 3.1. This lemma relates paths in the directed graph $G(A)$ and the nonzero elements that appear in the factors L and U, represented in the so-called filled graph $G^+(A)$.

LEMMA 3.1 (fill path (Rose and Tarjan [20])). *Let G be a directed or undirected graph whose vertices are the integers 1 through n, and let G^+ be its filled graph. Then $\langle i, j\rangle$ is an edge of G^+ if and only if there is a path in G from i to j whose intermediate vertices are all smaller than $\min(i, j)$.*

The filled graph $G^+(A)$ represents a symbolic bound for the factors L and U; that is, it ignores possible numeric cancellation during the factorization. The next lemma represents an example of conditions under which this structure prediction is exact, by taking into account the values of the nonzeros in the matrix. In this lemma, a square Hall submatrix of A denotes a square submatrix of A which satisfies the Hall property and which is formed by a subset of rows and columns of A that can be different and noncontiguous.

LEMMA 3.2 (Gilbert and Ng [11]). *Suppose A is square and nonsingular and has a triangular factorization $A = LU$ without pivoting. Suppose also that all of the diagonal elements of A, except possibly the last one, are nonzero and that every square Hall submatrix of A is nonsingular. Then $G(L + U) = G^+(A)$; that is, every nonzero predicted by the filled graph of A is actually nonzero in the factorization.*

We are interested in fill when the diagonal may contain zeros (perhaps due to pivoting), but Lemma 3.2 does not hold in this case. An example showing this was given by Brayton, Gustavson, and Willoughby [1]. We give a slightly different example in Figure 3.1, where we display a matrix A, its bipartite graph $H(A)$, and its directed graph $G(A)$. Note that $H(A)$ satisfies the strong Hall property. Since there is a path from 5 to 4 through lower numbered vertices in $G(A)$, the edge $\langle 5, 4\rangle$ belongs to the filled graph $G^+(A)$, but $L_{54} = 0$ regardless of the nonzero values of A. That is because after the first step of elimination, the elements in column positions 2 and 4 of the rows 2 and 5 are linearly dependent. At the second step of elimination the element L_{54} is zeroed.

A simpler way of understanding this numeric cancellation is to consider the two submatrices $A([1\colon 3, 5], 1\colon 4)$ and $A(1\colon 3, 1\colon 3)$ and their determinants that determine

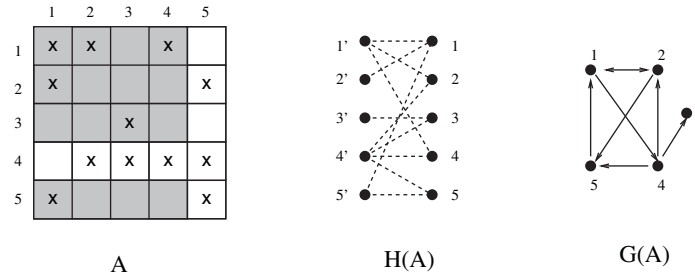


FIG. 3.1. *Example showing that the fill path Lemma* 3.2 *does not predict exactly the nonzero structure of* L *and* U *when factorizing without pivoting the strong Hall matrix* A*. Details are given in the text following Lemma* 3.1.

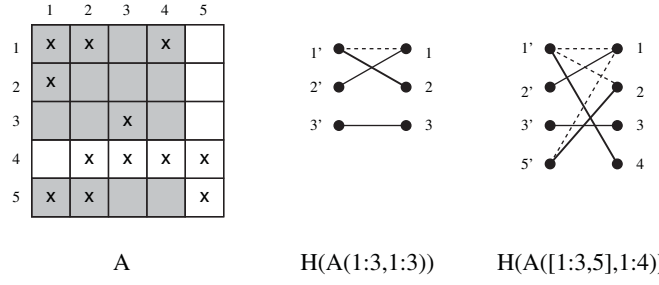


FIG. 3.2. *Example for Theorem* 3.4 $(1 \to 2)$*. Consider the strong Hall matrix* A *and the matrix* $B = A([1\colon 3, 5], 1\colon 4)$ *displayed in patterned gray. The Hall set of maximum cardinality of* $H(A([1\colon 3, 5], 1\colon 3))$ *is* $C_{35} = \{3\}$ *and* $R_{35} = \{3'\}$*. Suppose element* L_{54} *is nonzero. The perfect matching* M_k *of matrix* $A_k = A(1\colon 3, 1\colon 3)$ *is formed by the edges* $\langle 1', 2\rangle$*,* $\langle 2', 1\rangle$*, and* $\langle 3', 3\rangle$*. The perfect matching* M_B *of* B *is formed by the edges* $\langle 1', 4\rangle$*,* $\langle 2', 1\rangle$*,* $\langle 3', 3\rangle$*, and* $\langle 5', 2\rangle$*. Form a path by starting at* $5'$ *and by following one edge in* M_B *and one edge in* M_k*. This yields the path* $(5', 2, 1', 4)$*.*

the value of L_{54}. The submatrix $A([1\colon 3, 5], 1\colon 4)$ (displayed in light gray in Figure 3.1) has three columns (2, 3, and 4) with nonzero elements in only two rows (1 and 3). This submatrix does not satisfy the Hall property, and its determinant is zero. This is the approach we use to identify some numeric cancellations in the LU factorization.

The following lemma describes the above observation. Assuming that the LU factorization exists, this lemma relates the value of an element of the factors L and U to the singularity of a submatrix of A.

LEMMA 3.3 (Gilbert and Ng [11]). *Suppose* A *is square and nonsingular and has a triangular factorization* $A = LU$ *without pivoting. Let* i *be a row index and* j *a column index of* A*, and let* B *be the submatrix of* A *consisting of rows* 1 *through* $\min(i, j) - 1$ *and* i*, and columns* 1 *through* $\min(i, j) - 1$ *and* j*. Then* $(L + U)_{ij}$ *is zero if and only if* B *is singular.*

3.1. New results. Theorem 3.4 is the first new result of this section and provides necessary and sufficient conditions, in terms of paths in the bipartite graph $H(A)$ for a fill element to occur in exact arithmetic during Gaussian elimination. It is illustrated in Figures 3.2 and 3.3. Consider the nonzero structure of L. Suppose that the factorization exists until the step $j - 1$ of factorization; that is, the principal minor of order $j - 1$ is nonzero. The theorem uses the fact that L_{ij} is nonzero if and only if the determinant of the submatrix $A([1\colon j - 1, i], 1\colon j)$ is nonzero.

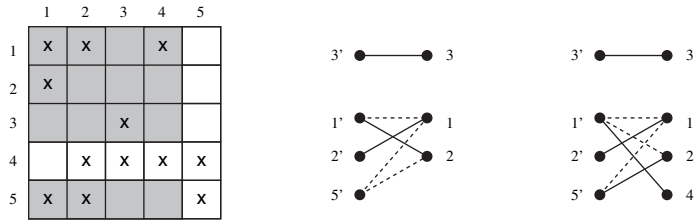


FIG. 3.3. *Example for Theorem* 3.4 ($3 \rightarrow 1$). *Consider the strong Hall matrix* A *as in Figure* 3.2, *the matrix* $B = A([1\colon 3, 5], 1\colon 4)$, $C_{35} = \{3\}$, *and* $R_{35} = \{3'\}$. *Consider the path* $\mathcal{Q} = (5', 1, 1', 4)$ *that has no vertex in* $C_{35} \cup R_{35}$. *The graph* $H(A([1\colon 3, 5], 1\colon 3))$ *is partitioned into two subgraphs. The subgraph induced by column vertex* 3 *and row vertex* $3'$ *satisfies the Hall property, and it has a perfect matching* $\widetilde{M} = \{\langle 3', 3\rangle\}$. *The subgraph induced by column vertices* $\{1, 2\}$ *and row vertices* $\{1', 2', 5'\}$ *satisfies the strong Hall property and has a column-complete matching* $\widehat{M} = \{\langle 1', 2\rangle, \langle 2', 1\rangle\}$ *for which there is a c-alternating path* $\mathcal{R} = (1', 2, 5')$. *The matching* M *is formed by the edges of* $\widetilde{M}$ *and* $\widehat{M}$ *and is presented by solid edges in the graph* $H(A([1\colon 3, 5], 1\colon 3))$. *The matching obtained by alternating along path* $\mathcal{R}$ *is a perfect matching of* $H(B)$ *and is presented at the right of the figure.*

THEOREM 3.4. *Let* A *be an* $n \times n$ *nonsingular matrix that has a triangular factorization* $A = LU$. *Suppose that every square Hall submatrix of* A *is nonsingular. Let* i *be a row of* A, j *be a column of* A, *and* $k = \min(i, j) - 1$. *Let* M_k *be a perfect matching of* $A(1\colon k, 1\colon k)$. *Let* C_{ki} *be the Hall set of maximum cardinality in* $H(A([1\colon k, i], 1\colon k))$ *and* R_{ki} *be the set of all row indices covered by columns of* C_{ki}. *Then the following three statements are equivalent:*

1. $(L + U)_{ij}$ *is nonzero.*
2. *There is an r-alternating path in the bipartite graph* $H(A)$ *from row vertex* i' *to column vertex* j *with respect to the matching* M_k.
3. *There is a path in the bipartite graph* $H(A)$ *from* i' *to* j *whose intermediate vertices are smaller than or equal to* k *and that has no vertex in* $C_{ki} \cup R_{ki}$.

Proof. Let A_k be the leading $(k \times k)$ principal submatrix of A and $\det(A_k)$ be its determinant. As we suppose the factorization exists, $\det(A_k)$ is nonzero. This implies that A_k satisfies the Hall property and has a perfect matching M_k. The matching M_k also represents a column-complete matching in the graph $H(A([1\colon k, i], 1\colon k))$. Lemma 2.7 applies with respect to the graph $H(A([1\colon k, i], 1\colon k))$ and the Hall set C_{ki} and says that each row vertex of R_{ki} has to be matched by M_k to one of the column vertices in C_{ki}. Since i' is not a row vertex matched by M_k, then $i' \notin R_{ki}$.

Let B be the submatrix of A consisting of columns 1 through k and j and rows 1 through k and row i. Suppose edge $\langle i', j\rangle$ does not belong to $H(A)$; otherwise the proof is trivial. We will prove now that the three statements are equivalent.

$1 \rightarrow 2$. As $(L+U)_{ij}$ is nonzero by hypothesis, then Lemma 3.3 applies and shows that B is a nonsingular matrix. Hence its bipartite graph $H(B)$ satisfies the Hall property, and there is a perfect matching M_B in $H(B)$.

Consider now the row vertex i' in the bipartite graph $H(A)$. Recall we assume that edge $\langle i', j\rangle$ does not belong to $H(A)$. Row vertex i' is matched by M_B to column vertex j_0. Since $i' \notin R_{ki}$, we can deduce that $j_0 \notin C_{ki}$. Column vertex j_0 is matched by M_k to some row vertex i'_0, where $i'_0 \neq i'$ since i' is not matched by M_k. Also we have that $i'_0 \notin R_{ki}$. Row vertex i'_0 is matched by M_B to some column vertex j_1, where $j_1 \neq j_0$ since j_0 is matched in M_B to i'. If $j_1 = j$, then we stop. Otherwise, we continue our reasoning. For each row vertex we consider its matched column vertex by M_B;

then for each column vertex we consider its matched row vertex by M_k. Continuing inductively, we arrive at vertex j. The vertices followed during our reasoning are vertices $i', j_0, i'_0, j_1, i'_1, \ldots, i'_t, j$. Edge $\langle i', j_0\rangle$ and edges $\langle i'_q, j_{q+1}\rangle$ are edges of $H(B)$ which belong to the perfect matching M_B. Edge $\langle i'_t, j\rangle$ and edges $\langle j_q, i'_q\rangle$ are edges of $H(A_k)$ which belong to the perfect matching M_k. This yields a path in $H(A)$ from row vertex i' to column vertex j that is r-alternating with respect to the matching M_k.

$2 \rightarrow 3$. Consider the r-alternating path $(i', j_0, i'_0, j_1, i'_1, \ldots, i'_t, j)$ from i' to j with respect to the matching M_k. All of the intermediate vertices on this path are smaller than or equal to k. Because $i' \notin R_{ki}$, we can deduce that $j_0 \notin C_{ki}$. Continuing inductively, we can deduce that this path does not include any vertex in $C_{ki} \cup R_{ki}$.

$3 \rightarrow 1$. Let d' be the last row vertex on $\mathcal{Q}$, that is, the vertex just before j on $\mathcal{Q}$. We partition the graph $H(A([1\colon k, i], 1\colon k))$ into two subgraphs. The first subgraph, induced by the row vertices in R_{ki} and the column vertices in C_{ki}, satisfies the Hall property and has a perfect matching $\widehat{M}$. The second subgraph, induced by the row vertices $1, \ldots k'$ and i', except row vertices in R_{ki}, and the column vertices 1 through k, except column vertices in C_{ki}, is strong Hall. Lemma 2.6 says that there is a column-complete matching $\widehat{M}$ which excludes row vertex i' and for which there exists a c-alternating path $\mathcal{R}$ from d' to i'.

Let the matching M be formed by the edges of $\widehat{M}$ and the edges of $\widetilde{M}$. This matching represents a column-complete matching in $H(A([1\colon k, i], 1\colon k))$. We now show that the graph $H(B)$ satisfies the Hall property. Recall that column vertex j and row vertex i' are not matched by the matching M. Consider path $\mathcal{R}$ from d' to i' that is c-alternating with respect to matching M. Obtain a new matching $M \oplus \mathcal{R}$ from M by alternating along path $\mathcal{R}$. As i' is not matched in M and d' is matched in M, then i' is matched in $M \oplus \mathcal{R}$ and d' is not matched in $M \oplus \mathcal{R}$. Add to matching $M \oplus \mathcal{R}$ the edge $\langle d', j\rangle$.

Thus we obtain a perfect matching in $H(B)$; that is, $H(B)$ satisfies the Hall property. By hypothesis, every square Hall submatrix of A is nonsingular, and thus B is nonsingular and its determinant is nonzero. Therefore $(L+U)_{ij}$ is nonzero. ∎

The next theorem uses Hall sets of maximum cardinality associated with subsets of columns of A to restrict paths corresponding to nonzero elements of L and U. In this paper we use this theorem in section 4 to determine upper bounds for the factorization $PA = LU$, where the matrix A satisfies only the Hall property. Note that for a matrix satisfying the strong Hall property, the Hall set of maximum cardinality of a subset of columns is always empty. Thus Theorem 3.5 is relevant to matrices satisfying only the Hall property. This theorem can also be useful in the algorithm described in section 3.2. The Hall sets involved can be computed prior to the factorization using an algorithm as, for example, the one proposed in [15].

THEOREM 3.5. *Let A be an $n \times n$ nonsingular matrix that is factored by Gaussian elimination as $A = LU$. Suppose that $(L+U)_{ij}$ is nonzero. Let $k = min(i, j) - 1$, and let C_k be the Hall set of maximum cardinality in the first k columns and R_k be the set of all row indices covered by columns of C_k. Then there is a path in the bipartite graph $H(A)$ from row vertex i' to column vertex j whose intermediate vertices are smaller than or equal to k and that has no vertex in $C_k \cup R_k$.*

Proof. Let C_{ki} be the Hall set of maximum cardinality in $H(A([1\colon k, i], 1\colon k))$ and R_{ki} be the set of all row indices covered by columns of C_{ki}. It can be easily shown that $C_k \subseteq C_{ki}$ and $R_k \subseteq R_{ki}$. The third statement of Theorem 3.4 implies that this theorem holds. ∎

Note that Theorem 3.5 provides only a necessary condition for fill to occur during the elimination. Figure 3.4 (as well as Theorem 3.4) shows that the condition is

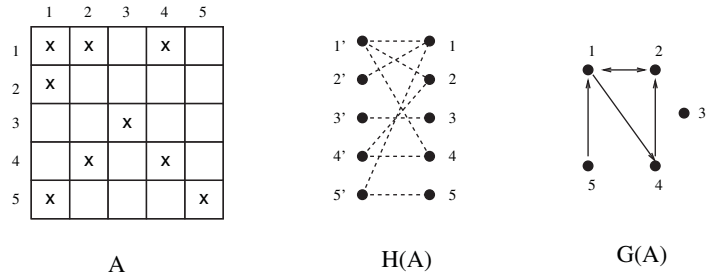


FIG. 3.4. *Example showing that the converse of Theorem* 3.5 *is not true.*

not sufficient. Consider the Hall matrix A in Figure 3.4. The Hall set of maximum cardinality is $C_3 = \{3\}$, and it covers the row index $R_3 = \{3'\}$. The Hall set of maximum cardinality in $H(A([1\!:\!3, 5], 1\!:\!3))$ is $C_{35} = \{2, 3\}$, and it covers the row indices $R_{35} = \{1', 3'\}$. There is a path $(5', 1, 1', 4)$ in $H(A)$ that has no vertex in C_3. However, the element $L_{54} = 0$ because of numeric cancellation.

3.2. Computing the nonzero structure of the factors L and U during Gaussian elimination with partial pivoting. In this section we present an algorithm that uses the results of the previous section to compute the nonzero structure of the factors L and U during the LU factorization with partial pivoting. The algorithm computes one column of L and one row of U at a time.

First, we present Theorem 3.6 that describes explicitly how Theorem 3.4 can be used during the LU factorization with partial pivoting of a matrix A. This theorem supposes that the first $j-1$ steps of the LU factorization exist, and it gives the necessary results to compute the structure of column j of L and of row j of U at the jth step of factorization.

THEOREM 3.6. *Let A be an $n \times n$ nonsingular matrix that is to be decomposed using LU factorization with partial pivoting. Suppose that the first $j-1$ steps of LU factorization with partial pivoting of A exist and have been executed. Let $P_{J-1} = P_{j-1}P_{j-2}\dots P_1$ be the permutations performed during the first $j-1$ steps of elimination, and let M_{j-1} be a perfect matching of $(P_{J-1}A)(1\!:\!j-1, 1\!:\!j-1)$. Suppose that every square Hall submatrix of A is nonsingular. At the jth step of decomposition, the element L_{ij} is nonzero if and only if there is a c-alternating path in the bipartite graph $H(P_{J-1}A)$ from column vertex j to row vertex i' with respect to the matching M_{j-1}. The element U_{ji} is nonzero if and only if there is an r-alternating path in the bipartite graph $H(A)$ from row vertex j' to column vertex i with respect to the matching M_{j-1}.*

Proof. The proof is similar to the proof of Theorem 3.4. □

Algorithm 1 uses Theorem 3.6 and sketches the factorization $PA = LU$, where $P = P_{n-1}\dots P_1$ and each P_j reflects the permutation of two rows at step j of factorization. At each step j, the structure of column j of L is determined, and then its numerical values are computed. The element of maximum magnitude in column j of L is chosen as the pivot. Let L_{kj} be this element. The algorithm interchanges rows k and j of L and rows k and j of A. Then the structure of row j of U is determined, followed by the computation of its numerical values.

The structure of column j of L is computed by finding all of the c-alternating paths with respect to M_{j-1} from column vertex j to some row vertex i'. This can be achieved in a similar way to the augmenting path technique, used in finding maximum matchings in bipartite graphs and described, for example, in [7]. This technique

ensures that each edge of the bipartite graph of A is traversed at most once. The structure of row i of U is computed in a similar way. Since the jth diagonal element corresponds to a nonzero, Theorem 3.4 ensures that there is a c-alternating path $\mathcal{Q}$ from column vertex j to row vertex j' with respect to the matching M_{j-1}. During the computation of the structure of column j of L, we store for each row vertex i' the column vertex just before i' on a c-alternating path with respect to M_{j-1} from j to i'. This allows us to retrace $\mathcal{Q}$. The algorithm computes a new matching M_j by alternating along path $\mathcal{Q}$.

The overall complexity of computing the structure of L and the structure of U in Algorithm 1 is hence bounded by $O(n \cdot nnz(A))$, where n is the order and $nnz(A)$ is the number of nonzeros of matrix A.

ALGORITHM 1. LU factorization with partial pivoting, aware of some cancellations

$M_0 = \emptyset$
for $j := 1$ to n **do**
  if $j < n$ **then**
    1. Compute structure of $L(j\colon n, j)$. This is formed by all row vertices $i' \geq j$ such that there is a c-alternating path in $H(A)$ with respect to M_{j-1} from column vertex j to row vertex i'.
    2. Compute numerical values of $L(j\colon n, j)$.
    3. Find k such that $|L_{kj}| = \max |L(j\colon n, j)|$. Let $v = L_{kj}$.
    4. Interchange $L(j, :)$ with $L(k, :)$ and $A(j, :)$ with $A(k, :)$. Let $\mathcal{Q}[j\colon j']$ be the c-alternating path in $H(A)$ with respect to M_{j-1} that corresponds to L_{jj}.
    5. Scale: $L(:, j) = L(:, j)/v$.
  end if
  6. Compute structure of $U(j, j+1\colon n)$. This is formed by all column vertices $i \geq j$ such that there is an r-alternating path in $H(A)$ from row vertex j' to column vertex i with respect to the matching M_{j-1}.
  7. Compute numerical values of $U(j, j+1\colon n)$. Let $U_{jj} = v$.
  if $j = 1$ **then**
    $M_1 = \mathcal{Q}$
  else
    $M_j = M_{j-1} \oplus \mathcal{Q}$
  end if
end for

Several aspects need to be investigated and remain as open questions. The first important aspect is related to the practical interest of using this algorithm, which depends on the utility of identifying numeric cancellations and on the number of numeric cancellations that appear in real world applications. The second aspect is related to the complexity of Algorithm 1, which is equivalent to the complexity of one of the first algorithms for computing the structure of the factors L and U, denoted as the FILL2 algorithm in [20]. The algorithms proposed more recently for computing fill-ins [10] are faster in practice than FILL2. Since we expect Algorithm 1 to have a similar run time to FILL2, further investigation is required to make it competitive with respect to the new algorithms.

4. Tight exact bounds for the structure prediction of $PA = LU$, when A satisfies only the Hall property. Let A be an $n \times n$ matrix that satisfies the Hall property. Suppose A is factored by Gaussian elimination with row interchanges

as $PA = LU$. In this section we discuss the problem of predicting bounds for the factors L and U prior to the numerical factorization. We consider exact results; that is, the upper bounds do not include elements that correspond to numeric cancellations due to submatrices of A structurally singular.

The next three theorems give tight exact bounds for the nonzero structure of the factors L and U. Theorem 4.1 gives upper bounds for the structure of L and U in terms of paths in the bipartite graph $H(A)$. Theorems 4.2 and 4.3 show that this bound is the tightest possible for Gaussian elimination with row interchanges of a matrix that satisfies the Hall property. That is, for every predicted element of the upper bound, there is a permutation and a choice of the values of matrix A such that this element corresponds to a nonzero in the factors L or U.

THEOREM 4.1. *Let A be an $n \times n$ nonsingular matrix that is factored by Gaussian elimination with row interchanges as $PA = LU$. Let i be an index, j be a column index, and $q = \min(i, j) - 1$. Let C_q be the Hall set of maximum cardinality in the first q columns and R_q be the set of all row indices covered by columns of C_q. If L_{ij} is nonzero, then there is a path in the bipartite graph $H(A)$ from row vertex k' to column vertex j whose intermediate column vertices are all in $\{1, \ldots, q\}$ and that has no vertex in $C_q \cup R_q$, where k is the row of A that corresponds to row i of PA. If U_{ij} is nonzero, then there is a path in the bipartite graph $H(A)$ from column vertex i to column vertex j whose intermediate column vertices are all in $\{1, \ldots, q\}$ and that has no vertex in $C_q \cup R_q$.*

Proof of Case 1 ($i \geq j$ (*structure of* L)). Due to Theorem 3.5, there is a path $\mathcal{Q}$ in $H(A)$ from row vertex k' to column vertex j whose intermediate column vertices are all in $\{1, \ldots, j-1\}$ and that has no vertex in $C_{j-1} \cup R_{j-1}$. This is the path searched in the theorem. □

Proof of Case 2 ($i < j$ (*structure of* U)). According to Theorem 3.5, there is a path $\mathcal{Q}$ in $H(A)$ from row vertex k' to column vertex j whose intermediate column vertices are all in $\{1, \ldots, i-1\}$ and that has no vertex in $C_{i-1} \cup R_{i-1}$.

By hypothesis, the factorization exists; thus the ith diagonal element of PA is nonzero. Theorem 3.5 applies with respect to this element and says that there is a path $\mathcal{R}$ in $H(A)$ from column vertex i to row vertex k' whose intermediate column vertices are all in $\{1, \ldots, i-1\}$ and that has no vertex in $C_{i-1} \cup R_{i-1}$.

Using the path $\mathcal{R}$ and the path $\mathcal{Q}$, we can form a path in $H(A)$ from column vertex i to column vertex j whose intermediate column vertices are all in $\{1, \ldots, i-1\}$ and that has no vertex in $C_{i-1} \cup R_{i-1}$. This is the path searched in the theorem. □

The next two theorems show that the upper bound defined in Theorem 4.1 for the structure of L and U is tight. First, Theorem 4.2 shows that the bound for the structure of L is tight, and it is illustrated in Figure 4.1. Second, Theorem 4.3 shows that the bound for U is tight, and it is illustrated in Figure 4.2.

The bound for L depends on the row permutations of A. It considers every row i of the original matrix A. The bound identifies all column indices j that correspond to elements of row i that can become potentially nonzeros during the factorization through permutations. The bound for U is independent of row permutations of A. It identifies potential nonzeros U_{ij} using paths that relate column vertex i to column vertex j in the bipartite graph of A. None of the results assumes that the input matrix A has a zero-free diagonal.

THEOREM 4.2. *Let H be the structure of a square Hall matrix. Let j be a column vertex, C_{j-1} be the Hall set of maximum cardinality in the first $j-1$ columns, R_{j-1} be the set of row indices covered by columns in C_{j-1}, and i' be any row vertex not in R_{j-1}. Suppose that H contains a path from i' to j whose intermediate column*

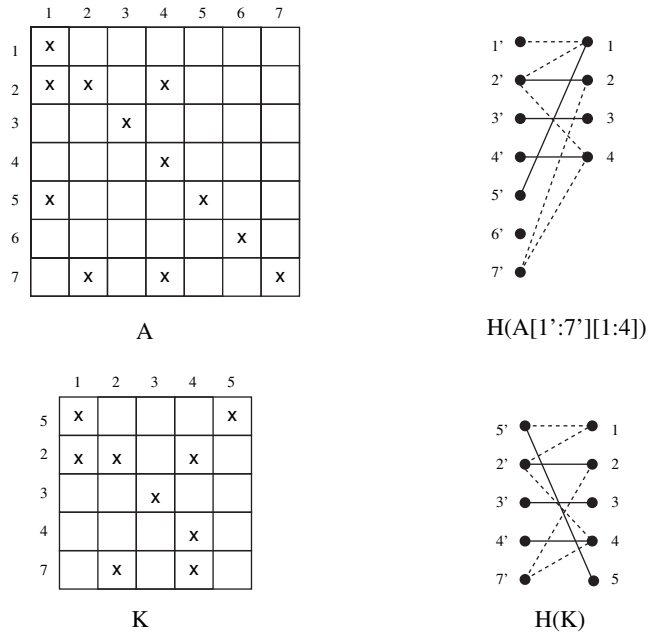


FIG. 4.1. *Example for Theorem* 4.2 *showing the construction that makes element* L_{75} *nonzero for the Hall matrix* A *at the top left. The last row vertex on the path* $(7', 4, 2', 1, 5', 5)$ *between* $i' = 7'$ *and* $j = 5$ *satisfying the conditions in Theorem* 4.2 *is* $e' = 5'$. *The graph* H_{j-1}, *presented at the top right, is the subgraph of* H *induced by column vertices* 1 *through* $j-1 = 4$ *and all of the row vertices. The solid edges represent a column-complete matching* M_{j-1} *that excludes row vertex* $7'$ *and with respect to which there is a c-alternating path* $\mathcal{R} = (5', 1, 2', 2, 7')$ *from* $5'$ *to* $7'$. *At the bottom right,* K *is the submatrix of* PA *with columns* 1 *through* $j = 5$ *and the rows in corresponding positions after four steps of pivoting. The fifth row of* K *is* $k' = i' = 7'$. *In* $H(K)$ *there is a maximum matching* $M_{j-1} \oplus \mathcal{R}$ *represented by solid edges at the bottom right. Thus the element* L_{75} *is nonzero.*

vertices are all in $\{1, \ldots, j-1\}$ *and which has no vertex in* $C_{j-1} \cup R_{j-1}$. *There exists a nonsingular matrix* A *with* $H(A) = H$ *and a permutation matrix* P *such that if* A *is factored by Gaussian elimination with row interchanges as* $PA = LU$, *then row* i *of* A *is permuted in some row position* k *of* PA, $k \geq j$ *and* $L_{kj} \neq 0$.

Proof. By hypothesis, there is a path in H from i' to j whose intermediate column vertices are all at most j. Consider H_{j-1} the subgraph of H induced by all row vertices and all column vertices from 1 to $j-1$. The graph H satisfies the Hall property, and hence H_{j-1} also satisfies the Hall property. We obtain a column-complete matching M_{j-1} in this graph which will induce the pivoting order for the first $j-1$ steps of elimination. We partition the graph H_{j-1} into two subgraphs. The first subgraph $\widetilde{H}_{j-1}$ satisfies the Hall property and is induced by all of the row vertices in R_{j-1} and all of the column vertices in C_{j-1}. Let $\widetilde{M}_{j-1}$ be a perfect matching in this subgraph. The second subgraph $\widehat{H}_{j-1}$ satisfies the strong Hall property and is induced by all of the row vertices except row vertices in R_{j-1} and all of the column vertices 1 through $j-1$ except column vertices in C_{j-1}. Let $\widehat{M}_{j-1}$ be a column-complete matching in this subgraph.

We distinguish two cases to determine $\widehat{M}_{j-1}$, depending on if $\langle i', j\rangle$ is an edge of $H(A)$ or not. First, assume that $\langle i', j\rangle$ is an edge of $H(A)$. Lemma 2.4 says that for any column-complete matching M of $\widehat{H}_{j-1}$ there is a c-alternating path $\mathcal{R}$ from i' to some unmatched row vertex. We denote by $\widehat{M}_{j-1}$ the matching obtained from

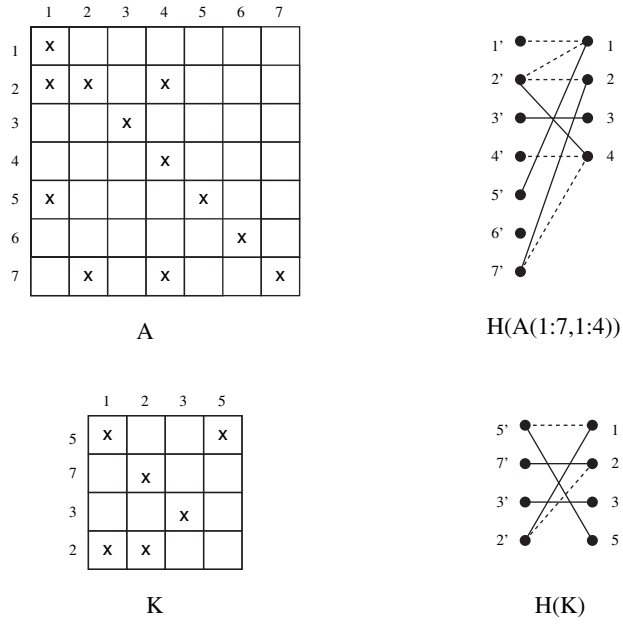


FIG. 4.2. *Example for Theorem* 4.3 *showing the construction that makes element* U_{45} *nonzero for the Hall matrix* A *at the top left. The Hall set of maximum cardinality in the first three columns is* $C_3 = \{3\}$, $R_3 = \{3'\}$. *Consider the path* $\mathcal{Q} = (5, 5', 1, 2', 4)$ *satisfying the conditions in Theorem* 4.3 ($k' = 5'$ *and* $e' = 2'$). *The graph* $\widehat{H}_4$, *presented at the top right, is the subgraph of* H *induced by column vertices* 1 *through* $i = 4$ *and all of the row vertices. The solid edges represent a column complete-matching* M_4 *that is formed by the edge* $\langle 2', 4\rangle$, *the matching* $\widehat{M}_3$ *(formed by the edges* $\langle 5', 1\rangle, \langle 7', 2\rangle, \langle 2', 4\rangle$*), and the matching* $\widetilde{M}_3$ *(formed by the edge* $\langle 3', 3\rangle$*). This matching determines the pivoting order for the first four steps of elimination. With respect to the matching* $\widehat{M}_3$ *there is a c-alternating path* $\mathcal{R} = (5', 1, 2')$. *Consider the matrix* $K = A([5', 7', 3', 2'][1\colon 3, 5])$ *presented at the bottom left and its graph presented at the bottom right. The perfect matching* M *is presented by solid edges in the graph* $H(K)$. *Since* K *satisfies the Hall property and the minor of order-*3 *of* PA *is nonzero, then the element* U_{45} *is nonzero.*

M by alternating along path $\mathcal{R}$. With this choice, row vertex i' is not covered by the matching $\widehat{M}_{j-1}$. Second, assume that $\langle i', j\rangle$ is not an edge of $H(A)$. Let e' be the last row vertex on the path between i' and j, that is, the vertex just before j. Therefore Lemma 2.6 applies and says that there is a column-complete matching $\widehat{M}_{j-1}$ which excludes vertex i' and for which there exists a c-alternating path $\mathcal{R}$ from e' to i'.

Let the column-complete matching M_{j-1} be formed by the edges of $\widehat{M}_{j-1}$ and the edges of $\widetilde{M}_{j-1}$. We choose the values of A such that every square submatrix of A that is Hall, including A itself, is nonsingular. We can say that this is possible by using an argument as the one described in [11] (the determinant of a Hall submatrix is a polynomial in its nonzero values, not identically zero, since the Hall property implies a perfect matching). We choose the values of the nonzeros of A corresponding to edges of M_{j-1} to be larger than n and the values of the other nonzeros of A to be between 0 and 1. With this choice, Lemma 2.1 says that the first $j-1$ steps of elimination of A pivot on nonzeros corresponding to edges of M_{j-1}. Let P be the permutation matrix that describes these row interchanges.

Note that with our choice of M_{j-1}, row vertex i' is not covered by the matching M_{j-1}. Thus, after the first $j-1$ steps of elimination, row i of A was moved to a row in position k of PA, where $k \geq j$. We prove that this choice makes L_{kj} nonzero. If $\langle i', j\rangle$

is an edge of $H(A)$, then L_{kj} is nonzero. Otherwise, let K be the $j \times j$ submatrix of A that includes the first j columns and the rows 1 to $j-1$ in corresponding positions of PA and row i of A (that is, row k of PA). Thus the columns of K are those numbered 1 through j in $H(A)$. The first $j-1$ columns are matched by M_{j-1}, while the last column j is not matched by M_{j-1}. The first $j-1$ rows of K are those matched to columns 1 through $j-1$ of $H(A)$ by M_{j-1}. The last row of K is row number i of A.

To show that L_{kj} is nonzero, we still need to show that K satisfies the Hall property. Recall that column vertex j and row vertex i' are not matched by the matching M_{j-1} in $H(K)$. Consider path $\mathcal{R}$ from e' to i' that is c-alternating with respect to matching M_{j-1}. Obtain a new matching $M_{j-1} \oplus \mathcal{R}$ from M_{j-1} by alternating along path $\mathcal{R}$. As i' is not matched in M_{j-1} and e' is matched in M_{j-1}, then i' is matched in $M_{j-1} \oplus \mathcal{R}$ and e' is not matched in $M_{j-1} \oplus \mathcal{R}$. Add to matching $M_{j-1} \oplus \mathcal{R}$ the edge $\langle e', j \rangle$, and thus we get a perfect matching in $H(K)$; that is, $H(K)$ satisfies the Hall property. By our choice of values, every submatrix that satisfies the Hall property is nonsingular. Therefore element L_{kj} is nonzero. □

THEOREM 4.3. *Let H be the structure of a square Hall matrix. Let i and j be two column vertices, $i < j$, let C_{i-1} be the Hall set of maximum cardinality in the first $i-1$ columns, and let R_{i-1} be the row vertices covered by columns in C_{i-1}. Suppose that H contains a path from j to i whose intermediate column vertices are all in $\{1, \ldots, i-1\}$ and that has no vertex in $C_{i-1} \cup R_{i-1}$. There exists a nonsingular matrix A with $H(A) = H$ and a permutation matrix P such that if A is factored by Gaussian elimination with row interchanges as $PA = LU$, then U_{ij} is nonzero.*

Proof. By hypothesis, there is a path $\mathcal{Q}$ in $H(A)$ from column vertex j to column vertex i whose intermediate column vertices are all at most $i-1$. Let k' be the first row vertex on $\mathcal{Q}$, that is, the vertex just after j on $\mathcal{Q}$. Let e' be the last row vertex on $\mathcal{Q}$, that is, the vertex just before i on $\mathcal{Q}$. Note that e' can be equal to k'.

Let $\widehat{H}_{i-1}$ be the strong Hall subgraph of H induced by all of the row vertices except row vertices in R_{i-1} and all of the column vertices 1 through $i-1$ except column vertices in C_{i-1}. Lemma 2.6 says that there is a column-complete matching $\widehat{M}_{i-1}$ which excludes e' and for which there exists a c-alternating path $\mathcal{R}$ from k' to e'. (If $k' = e'$, then $\mathcal{R}$ is empty.) Let $\widetilde{H}_{i-1}$ be the subgraph of $H(A)$ induced by all of the row vertices in R_{i-1} and all of the column vertices in C_{i-1}. The graph $\widetilde{H}_{i-1}$ satisfies the Hall property, and Lemma 2.6 says that there is a perfect matching $\widetilde{M}_{i-1}$ in $\widetilde{H}_{i-1}$.

Consider H_i the subgraph of H induced by all of the row vertices and all of the column vertices 1 through i. The matching M_i formed by the edge $\langle e', i \rangle$, all of the edges of $\widehat{M}_{i-1}$, and all of the edges of $\widetilde{M}_{i-1}$ is a column-complete matching in H_i . We choose the values of A such that every square submatrix of A that is Hall, including A itself, is nonsingular. We set the values of the nonzeros of A corresponding to edges of M_i to be larger than n and the values of the other nonzeros of A to be between 0 and 1. With this choice the first i steps of elimination of A pivot on nonzeros corresponding to edges of M_i (Lemma 2.1). Let P be the permutation matrix that describes these row interchanges.

We prove that this pivoting choice makes U_{ij} nonzero. Let K be the submatrix $PA(1{:}\,i, [1{:}\,i-1, j])$. To show that U_{ij} is nonzero, we need to show that the graph $H(K)$ satisfies the Hall property. For this, consider again the matching $\widehat{M}_{i-1}$ and the c-alternating path $\mathcal{R}$ from k' to e'. Consider the path formed by the edge $\langle j, k' \rangle$ followed by $\mathcal{R}$, and consider the matching M obtained by alternating along this path. Since k' is matched by $\widehat{M}_{i-1}$ and j is unmatched by $\widehat{M}_{i-1}$, then both k' and j are matched by M, and its cardinality is $|\widehat{M}_{i-1}| + 1$. We add to matching M the edges of

$\widetilde{M}_{i-1}$. Thus M is a perfect matching in $H(K)$; that is, this matrix satisfies the Hall property, and its determinant is nonzero. This shows that U_{ij} is nonzero. □

We make one final note on the similarities between the exact structure prediction presented in this section and the sparsity analysis of the QR factorization for square matrices satisfying the Hall property. The structure prediction for the QR factorization of matrices satisfying only the Hall property was studied by Hare et al. in [15] and Pothen in [19]. It can be easily shown that the structure of Q represents a tight exact bound for the structure of L of the factorization $PA = LU$ and that the structure of R is a tight exact bound for the structure of U obtained from Gaussian elimination with row interchanges.

5. The row merge graph and structure prediction for $A = P_1L_1 \dots P_{n-1}L_{n-1}U$. Let A be an $n \times n$ matrix with nonzero diagonal that satisfies the Hall property. Suppose A is factored by Gaussian elimination with row interchanges as $A = P_1L_1P_2L_2 \dots P_{n-1}L_{n-1}U$ and $\widetilde{L}$ is the union of the L_i. An upper bound for the nonzero structure of $\widetilde{L}$ and U was proposed by George and Ng [9]. This upper bound, called the row merge graph, contains the nonzeros in the factors for all possible row permutations that can later appear in the numerical factorization due to pivoting. In this section we discuss the row merge graph as an upper bound for the nonzero structure of the factors $\widetilde{L}$ and U when the matrix A satisfies only the Hall property. Thus we extend the work of Gilbert and Ng who showed in [11] that the row merge graph is a tight upper bound for Gaussian elimination with row permutations of strong Hall matrices.

First, we consider an exact analysis; that is, we assume only that the nonzero values in A are algebraically independent of each other. By a simple counterexample we show that for matrices satisfying only the Hall property, the row merge graph is not a tight bound for the factors $\widetilde{L}$ and U in the exact sense. This means that the row merge graph predicts as nonzero elements of $\widetilde{L}$ and U that during the actual factorization are zeroed. Second, we relax the condition on the numerical values of the nonzeros of A by considering a symbolic analysis. This is a weaker analysis than the exact analysis performed in section 4, since we ignore the possibility of numeric cancellation during the factorization. With this assumption, we show that the row merge graph is a tight bound for the factors $\widetilde{L}$ and U. In other words, for every edge of the row merge graph of a Hall matrix, there is a permutation such that this edge corresponds to a symbolic nonzero in the factors $\widetilde{L}$ or U.

5.1. Existing results. The row merge graph was proposed by George and Ng [9] as an upper bound for the nonzero structure of $\widetilde{L}$ and U and is obtained as follows: at each step of elimination an upper bound of the structure of $\widetilde{L}$ and U is computed. Consider step i and all of the rows that are candidates to pivoting at this step. An upper bound of their structure is given by the union of their structures. Thus the structure of each row candidate to pivoting is replaced by this union. The bipartite graph that contains all of the edges of the upper bound of $\widetilde{L}$ and U is called the *row merge graph*, denoted by $H^{\times}(A)$. The matrix containing a nonzero element for each edge of $H^{\times}(A)$ is referred to as the row merge matrix of A, denoted as $A^{\times}$. Several results in the literature use a directed version of the row merge graph, denoted as $G^{\times}(A)$ or $G^{\times}(H)$. This graph has n vertices and an edge for each nonzero of $A^{\times}$. The next theorem proves the claim that the row merge graph is an upper bound for the structure of $\widetilde{L}$ and U.

THEOREM 5.1 (George and Ng [9]). *Let A be a nonsingular square matrix with nonzero diagonal. Suppose Gaussian elimination with row interchanges is performed*

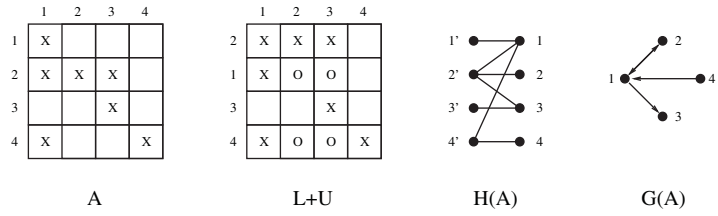


FIG. 5.1. *Example matrix A showing that the row merge graph is not an exact tight bound. The nonzero elements of A are denoted by* x*, and the fill elements of the factors $\widetilde{L}$ and U (corresponding to edges of $G^+(PA)$) are denoted by* o.

as $A = P_1 L_1 \dots P_{n-1} L_{n-1} U$, and let $\widetilde{L}$ be the union of the L_i. Then

$$G(L+U) \subseteq G^{\times}(A).$$

When the matrix satisfies the strong Hall property, Gilbert and Ng [11] showed that this graph represents a tight exact bound for the structure of $\widetilde{L}$ and U. That is, having a strong Hall graph H, for every edge $\langle i', j\rangle$ in its row merge graph $H^{\times}$, there exists a nonsingular matrix A (depending on i' and j) with $H(A) = H$ such that the element in position (i, j) of $L+U$ is nonzero. Nothing is known for the case when the matrix satisfies only the Hall property, and this question is the subject of this section.

5.2. The row merge graph and counterexample for tight exact bounds. In Figure 5.1 we give a counterexample showing that the row merge graph is not tight in the exact sense. The edge $\langle 4', 3\rangle$ is an edge of the row merge graph $H^{\times}(A)$. We present a permutation that makes the entry in position $(4, 3)$ nonzero in the factor $\widetilde{L}$. At the first step of elimination we pivot on the element at position $(2, 1)$, while at the next steps of elimination we pivot on the diagonal. Let P be the matrix describing these permutations. The directed graph $G(PA)$ has a path $(4, 1, 3)$; therefore the element in position $(4, 3)$ fills in. Then the $\langle 4, 3\rangle$ entry in $G^+(PA)$ is nonzero, but $\widetilde{L}_{43} = 0$, regardless of the nonzero values of A. Note that there is no choice of pivot at the first step of elimination that fills the element at position $(4, 3)$. We conclude that there is no permutation that makes the element $\widetilde{L}_{43}$ nonzero.

5.3. The row merge graph as a tight symbolic bound. We now discuss a symbolic analysis; that is, we ignore the possibility of numeric cancellation during the factorization. With this assumption, we show that the row merge graph is a tight bound for the factors $\widetilde{L}$ and U.

An example of the construction of the row merge matrix is presented in Figure 5.2. At the first step of elimination, rows 1, 4, and 5 are candidates to pivoting. The union of their structure is formed, and it replaces the structure of each one of these rows. This is repeated at each step on the trailing matrix.

Row merge fill elements refer to elements that are zero in the original matrix A but are nonzero in the row merge matrix $A^{\times}$. Similarly, *row merge fill edges* refer to those edges that don't belong to $H(A)$ but belong to the row merge graph $H^{\times}(A)$. The row merge fill edges in the row merge graph $H^{\times}(A)$ are related to paths in the bipartite graph $H(A)$ by Definition 5.2 and Theorem 5.3.

DEFINITION 5.2 (Gilbert and Ng [11]). *A path $\mathcal{Q} = (i', j_1, i'_1, j_2, i'_2, \dots, j_t, i'_t, j)$ in $H(A)$ is a row merge fill path for LU elimination with partial pivoting if either*

	1	2	3	4	5	6	7	8	9	10
1	x	o	x	o	o					
2		x	o	o	o	o		o		o
3			x	o	o	o		o	o	o
4	x	x	o	x	o	o	o	o	o	o
5	x	o	o	x	x	o	o	o	o	o
6						x	o	o	o	o
7				x	x	o	x	o	o	o
8								x	o	o
9			x	o	o	x	o	o	x	o
10		x	o	x	o	x	o	x	o	x

FIG. 5.2. *Example to illustrate the construction of a row merge matrix* $A^{\times}$, *Theorems* 5.4 *and* 5.5. *The nonzero elements of* A *are denoted by* x, *and the row merge fill elements are denoted by* o.

$t = 0$ *or the following conditions are satisfied:*

1. $j_k < j$ *and* $j_k \leq i'$ *for all* $1 \leq k \leq t$.
2. *Let* j_p *be the largest* j_k. *Then there is some* q *with* $p \leq q \leq t$, $j_p \leq i'_q \leq n$, *and the three paths* $\mathcal{Q}[i' : j_p]$, $\mathcal{Q}[j_p : i'_q]$, *and* $\mathcal{Q}[i'_q : j]$ *are also row merge fill paths in* $H(A)$.

The next theorem due to Gilbert and Ng gives a necessary and sufficient condition for fill to occur in the row merge graph $H^{\times}(A)$.

THEOREM 5.3 (Gilbert and Ng [11]). *For two vertices* i', j *of the bipartite graph* $H(A)$, *the edge* $\langle i', j \rangle$ *is an edge of* $H^{\times}(A)$ *if and only if there is a row merge fill path joining* i' *and* j *in* $H(A)$.

We present two algorithms that use Definition 5.2 to decompose a row merge fill path in paths and edges of the bipartite graph $H(A)$. Algorithm 2 decomposes the row merge fill path $\mathcal{Q}[i' : j]$ in subpaths by recursively applying Definition 5.2. The recursivity is stopped when a path is reduced to an edge. Its aim is to record for each intermediate column vertex j_p its corresponding row vertex of the middle path $\mathcal{Q}[j_p : i'_q]$ (that is, $MC[j_p] = i'_q$). Note that the vertices belonging to $\mathcal{Q}[i' : j]$ are distinct, and hence each intermediate vertex belongs to one and only one middle path.

ALGORITHM 2. Decomposition in subpaths

Input $\mathcal{Q} = (i', j_1, i'_1, \ldots, j_t, i'_t, j)$
Output MC **array updated**
if $t \neq 0$ **then**
1. decompose $\mathcal{Q}[i' : j]$ in $\mathcal{Q}[i' : j_p]$, $\mathcal{Q}[j_p : i'_q]$, and $\mathcal{Q}[i'_q : j]$ such that j_p is the largest j_k, where $1 \leq k \leq t$ and $j_p \leq i'_q \leq n$ and the three paths are also row merge fill paths (Definition 5.2).
2. $MC[j_p] = i'_q$.
3. decompose each of the three paths (which is not an edge) in sub-fill paths.
end if

Algorithm 3 decomposes the row merge fill path $\mathcal{Q}[i' : j]$ in an alternating sequence of edges and middle paths that we refer to as $ASEM$. It is easy to check that this algorithm returns the sequence $ASEM = \{\langle i', k_1 \rangle, \mathcal{Q}[k_1 : MC[k_1]], \langle MC[k_1], k_2 \rangle, \mathcal{Q}[k_2 : MC[k_2]], \ldots, \mathcal{Q}[k_u : MC[k_u]], \langle MC[k_u], j \rangle\}$, where $u \leq t$ and $k_1 = j_1$.

Consider an edge of the row merge graph $\langle i', j \rangle$ and its associated row merge fill path $\mathcal{Q} = (i', j_1, i'_1, \ldots, j_t, i'_t, j)$. We define a pivoting strategy relative to this path. At each elimination step k, if column vertex k is an intermediate vertex of

ALGORITHM 3. Decomposition in alternating sequence of edges and middle paths

Input $\mathcal{Q} = (i', j_1, i'_1, \ldots, j_t, i'_t, j)$
Output alternating sequence $ASEM$
if $t \neq 0$ **then**
1. decompose $\mathcal{Q}[i' \colon j]$ in $\mathcal{Q}[i' \colon j_p]$, $\mathcal{Q}[j_p \colon i'_q]$ and $\mathcal{Q}[i'_q \colon j]$ such that j_p is the largest j_k, $j_p \leq i'_q \leq n$, and the three paths are also row merge fill paths (Definition 5.2).
2. decompose $\mathcal{Q}[i' \colon j_p]$ in an alternating sequence and assign it to $ASEM$.
3. add the middle sub-fill path $\mathcal{Q}[j_p \colon i'_q]$ at the end of the sequence $ASEM$.
4. decompose $\mathcal{Q}[i'_q \colon j]$ in an alternating sequence and add it at the end of the sequence $ASEM$.
5. return the sequence $ASEM$.
else
6. return the edge $\langle i', j \rangle$.
end if

the path $\mathcal{Q}[i' \colon j]$, then we pivot on the element in position $(MC[k], k)$, and P_k is the elementary permutation matrix that describes this pivoting. If column vertex k is not an intermediate vertex of $\mathcal{Q}[i' \colon j]$, then we pivot on the diagonal element; that is, the elementary permutation matrix P_k is the identity. We call this strategy of pivoting the middle correspondent pivoting strategy with respect to the path $\mathcal{Q}[i' \colon j]$. In the next theorem we prove that such a strategy is valid; that is, the LU factorization exists in a symbolic sense.

LEMMA 5.4. *Let A be a square matrix with nonzero diagonal that satisfies the Hall property. Let $\langle i', j \rangle$ be an edge of the row merge graph $H^{\times}(A)$ and $\mathcal{Q}[i' \colon j]$ be its corresponding fill path in $H(A)$. Let $P = P_{n-1} \ldots P_2 P_1$ be the permutation matrix describing the middle correspondent pivoting strategy relative to $\mathcal{Q}[i' \colon j]$. Gaussian elimination $A = P_1 L_1 \ldots P_{n-1} L_{n-1} U$ exists in the symbolic sense.*

Proof. If the fill path $\mathcal{Q}[i' \colon j]$ corresponds to an edge of $H(A)$, then we choose P to be the identity matrix. As we assume the matrix A has a nonzero diagonal, the Gaussian elimination exists in the symbolic sense. In the rest of the proof, we assume that $\langle i', j \rangle$ is not an edge of $H(A)$.

As the case $j = 1$ is trivial, we will assume that $j > 1$. We will prove this by induction. At the first step of elimination, if row vertex $1'$ and column vertex 1 do not belong to $\mathcal{Q}[i' \colon j]$, then we pivot on the element in position $(1, 1)$. If the column vertex 1 belongs to $\mathcal{Q}[i' \colon j]$, then consider the fill path $\mathcal{Q}[1 \colon k']$, where $k' = MC[1]$ and $k' \geq 1$. We can see that $\mathcal{Q}[1 \colon k']$ is an edge of $H(A)$, and thus we can pivot on the element A_{k1}. Note that according to Definition 5.2, we cannot have that row vertex $1'$ belongs to $\mathcal{Q}[i' \colon j]$ and column vertex 1 does not belong to $\mathcal{Q}[i' \colon j]$.

Consider the kth step of elimination, where $k < n$. Suppose that at each elimination step prior to k, the middle correspondent pivoting strategy was valid; that is, the diagonal elements of the permuted matrix are nonzero. We show that at this step k we can apply the same pivoting strategy. Let P_{K-1} be the permutation matrix that describes the first $k-1$ row interchanges, that is, $P_{K-1} = P_{k-1} \ldots P_1$. Let A_k be the $k \times k$ principal submatrix of $P_{K-1}A$ that includes the first k columns and the rows in corresponding positions of $P_{K-1}A$. The columns of A_k are those numbered 1 through k in $H(A)$; the rows of A_k are those given by the permutation matrix P_{K-1}. We add to the matrix A_k all of the diagonal elements, except the last one, nonzero by our hypothesis. In the directed graph $G(A_k)$ we will number the vertices from 1 to k.

First, we will prove that the kth diagonal element of the permuted matrix $P_{K-1}A$ is nonzero. This corresponds to the last diagonal element of A_k. If k' is not an intermediate row vertex of the path $\mathcal{Q}[i'\colon j]$, then during the first $k-1$ steps of elimination row k was not permuted, and the last diagonal element of A_k is nonzero by our hypothesis. If k' is an intermediate row vertex on the path $\mathcal{Q}[i'\colon j]$, then row k was permuted during the first $k-1$ steps of elimination. We denoted by d the row that at the kth step of elimination is in position k of matrix A_k. We now trace the pivoting process to discover where row d comes from. Let k_1 be the middle correspondent vertex of k' ($MC[k_1] = k', k_1 \leq k'$). If $k_1 = k$, then $d' = k'$. Otherwise, according to our pivoting choice, the element in position (k', k_1) was used as the pivot at step k_1, and thus row k was interchanged with the row in position k_1. At this point, either row vertex k'_1 does not belong to $\mathcal{Q}[i'\colon j]$, and then $d = k_1$, or else it belongs, and then row k_1 was used as the pivot in some column k_2, where it was interchanged with some row $k_2 < k_1$. Extending the induction, we arrive at a row vertex $k'_q = d'$, which is not an intermediate row vertex of $\mathcal{Q}[i'\colon j]$. The vertices in $H(A)$ followed while tracing the pivoting process form the path $(k, k', k_1, k'_1, k_2, k'_2, \ldots, k_q, d')$. On this path, the edge $\langle k', k_1\rangle$ and the edges $\langle k'_p, k'_{p+1}\rangle$, with $1 \leq p < q$, correspond to diagonal elements of A_k. Hence this path can be transformed into the path $(k, k_1, \ldots, k_q, k)$ in $G(A_k)$. As $k > k_1 > \cdots > k_q$, according to Theorem 3.1 this path is a fill path in the directed graph $G(A_k)$, and the kth diagonal element of $P_{K-1}A$ corresponds to a symbolic nonzero.

Second, we show that at elimination step k we can apply the middle correspondent pivoting strategy. We distinguish two cases.

Case 1 (column vertex k is not an intermediate column vertex of $\mathcal{Q}[i'\colon j]$). We have just proved that the kth diagonal element of $P_{K-1}A$ is an edge of the filled graph $G^+(A_k)$. We use as the pivot the diagonal element.

Case 2 (column vertex k is an intermediate column vertex of $\mathcal{Q}[i'\colon j]$). Let e' be the middle path correspondent vertex of k, that is, $MC[k] = e'$ and $k \leq e'$. Let $\mathcal{Q}[e'\colon k]$ be the fill path between e' and k which is a subpath of our initial path $\mathcal{Q}[i'\colon j]$.

If $e' = k'$ (that is, $MC[k] = k'$), then row k' was not involved in any row permutation. We use as the pivot the diagonal element. If $e' > k$, then let K be the $(k+1) \times (k+1)$ submatrix of $P_{K-1}A$ that includes the first k columns and the rows in corresponding positions of $P_{K-1}A$ and column e and row e' of $P_{K-1}A$. We add to matrix K the first k diagonal elements, which correspond to symbolic nonzeros by our hypothesis. The vertices of the directed graph $G(K)$ are the vertices 1 through k and vertex e.

In the following, we want to show that $\langle e, k\rangle$ is an edge of the directed graph $G^+(K)$. If path $\mathcal{Q}[e'\colon k]$ is simply an edge, then $\langle e, k\rangle$ is an edge of $G^+(K)$. Otherwise, we decompose path $\mathcal{Q}[e'\colon k]$ into an alternating sequence of edges and middle paths using Algorithm 3. The following sequence is obtained: $\{\langle e', e_1\rangle, \mathcal{Q}[e_1\colon MC[e_1]], \langle MC[e_1], e_2\rangle, \mathcal{Q}[e_2\colon MC[e_2]], \ldots, \mathcal{Q}[e_q\colon MC[e_q]], \langle MC[e_q], k\rangle\}$. We can rewrite the sequence as a directed path from vertex e to vertex k of $G(K)$: $(e, e_1, e_2, \ldots, e_q, k)$. The intermediate vertices on this path are less than both e and k, because of the row merge fill paths Definition 5.2. Therefore $\langle e, k\rangle$ is an edge of $G^+(K)$, and thus it corresponds to a symbolic nonzero. This shows that we can choose as the pivot the element in position (e, k) at this step of elimination, and this ends our proof. □

The next theorem shows that the row merge graph represents a tight bound for the nonzero structure of $\widetilde{L}$ and U, in the symbolic sense. It is illustrated in Figures 5.3, 5.4, 5.5, and 5.6.

THEOREM 5.5. *Let A be a square matrix with nonzero diagonal that satisfies the Hall property. Let $\langle i', j\rangle$ be an edge of the row merge graph $H^{\times}(A)$. There is a*

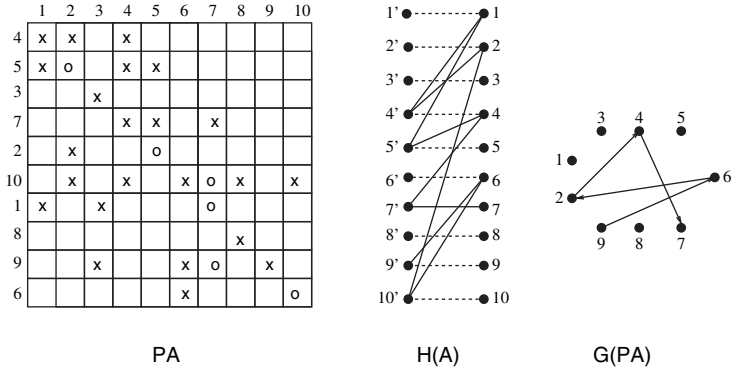


FIG. 5.3. *Example illustrating Theorem* 5.5 *and showing that* L_{97} *is nonzero for the row merge matrix presented in Figure* 5.2. *Consider the row merge fill path* $\mathcal{Q}[9'\colon 7] = (9', 6, 10', 2, 4', 1, 5', 4, 7', 7)$. *This path is displayed by solid edges in the bipartite graph* $H(A)$. *Figure* 5.4 *presents the decomposition of path* $\mathcal{Q}[9'\colon 7]$ *by Algorithms* 2 *and* 3. *First, Algorithm* 2 *decomposes* $\mathcal{Q}[9'\colon 7]$ *and obtains the following middle paths:* $\mathcal{Q}[6\colon 10'], \mathcal{Q}[2\colon 5'], \mathcal{Q}[4'\colon 1], \mathcal{Q}[4\colon 7']$. *This decomposition gives us the pivoting strategy, illustrated in the permuted matrix at the top left of Figure* 5.3. *Second, the fill path* $\mathcal{Q}[9'\colon 7]$ *is decomposed in an alternating sequence of edges and middle paths using Algorithm* 3. *This allows us to obtain the path* $(9, 6, 2, 4, 7)$ *which is a fill path in the directed graph of the permuted matrix* PA.

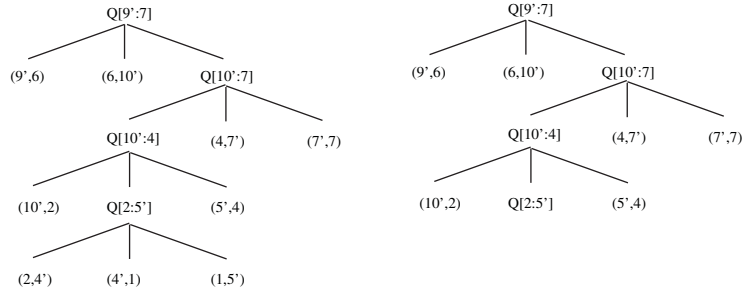


FIG. 5.4. *Example of the application of Algorithm* 2 *(left) and Algorithm* 3 *(right) on the path* $\mathcal{Q}[9'\colon 7] = (9', 6, 10', 2, 4', 1, 5', 4, 7', 7)$ *from Figure* 5.3.

permutation $P = P_1 P_2 \dots P_{n-1}$ *such that if* A *is factored by Gaussian elimination as* $A = P_1 L_1 \dots P_{n-1} L_{n-1} U$, *with* $\widetilde{L}$ *being the union of* L_i, *then* $(L + U)_{ij} \neq 0$, *in the symbolic sense.*

Proof. According to Theorem 5.3 there is a row merge fill path in $H(A)$ from row vertex i' to column vertex j. Let $\mathcal{Q}[i'\colon j]$ be formed by the vertices $(i', j_1, i'_1, \dots, j_t, i'_t, j)$. Assume that $t \neq 0$. At each step of elimination we pivot following the middle correspondent pivoting strategy with respect to the path $\mathcal{Q}[i'\colon j]$, as described in Lemma 5.4.

Assume now that we are at the jth step of elimination. Let P_{J-1} be the permutation matrix that describes the first $j-1$ row interchanges. Let K be the principal submatrix of $P_{J-1}A$ that includes the first j columns and column i and the rows in corresponding positions of PA (that is, if $i' \leq j$, then K is a $j \times j$ matrix; otherwise K is a $(j+1) \times (j+1)$ matrix). In matrix K we add diagonal elements, with $1 \leq i \leq j$, which are nonzero by our hypothesis. When $i > j$, we also add diagonal element (i', i) (row i was not permuted). The vertices of the directed graph $G(K)$ are numbered 1 through j and i.

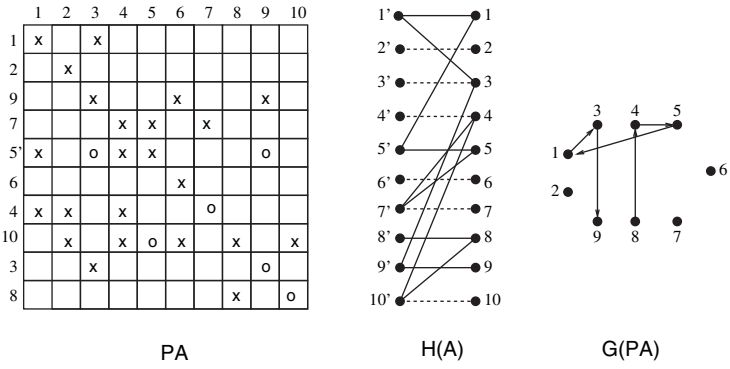


FIG. 5.5. *Example illustrating Theorem* 5.5 *and showing that* U_{89} *is nonzero for the row merge matrix presented in Figure* 5.2. *The row merge fill path* $Q[8':9] = (8', 8, 10', 4, 7', 5, 5', 1, 1', 3, 9', 9)$ *is displayed by solid edges in the bipartite graph* $H(A)$. *Figure* 5.6 *presents the decomposition of path* $Q[8':9]$ *by Algorithms* 2 *and* 3. *First, the path* $\mathcal{Q}[8':9]$ *is decomposed using Algorithm* 2, *and the following middle paths are obtained:* $\mathcal{Q}[8:10']$, $\mathcal{Q}[4:7']$, $\mathcal{Q}[5:5']$, $\mathcal{Q}[9':3]$, *and* $\mathcal{Q}[1:1']$. *This decomposition gives us the pivoting strategy, illustrated in the permuted matrix at the top left of Figure* 5.5. *Algorithm* 3 *decomposes the fill path* $\mathcal{Q}[8':9]$ *in an alternating sequence of edges and middle paths. This allows us to obtain the path* $(8, 4, 5, 1, 3, 9)$ *which is a fill path in the directed graph of the permuted matrix* PA.

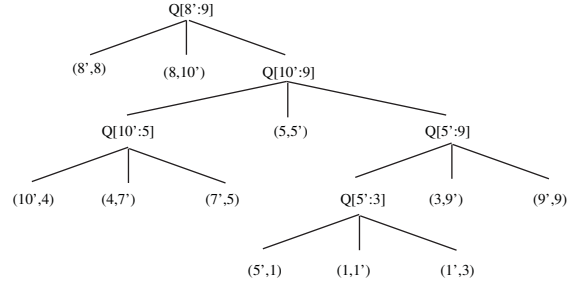


FIG. 5.6. *Example of the application of Algorithms* 2 *and* 3 *on the path* $\mathcal{Q}[8':9] = (8', 8, 10', 4, 7', 5, 5', 1, 1', 3, 9', 9)$ *from Figure* 5.5. *Both algorithms return the same result.*

Case 1 ($i > j$ (structure of $\widetilde{L}$)). The proof is similar to the proof of Lemma 5.4, in which a middle path becomes an edge of the filled graph of A, and we omit the details here.

Case 2 ($i < j$ (structure of U)). Consider row merge fill path $\mathcal{Q}[i':j] = (i', j_1, i'_1, \ldots, j_t, i'_t, j)$. We distinguish two cases. If column vertex i is not a vertex of path $\mathcal{Q}[i':j]$, then row i of A is not permuted during our pivoting strategy, and the proof is similar to the $\widetilde{L}$ case. If column vertex i is a vertex of path $\mathcal{Q}[i':j]$, Definition 5.2 decomposes this path in the following three paths: $\mathcal{Q}[i':i]$, $\mathcal{Q}[i:k']$, and $\mathcal{Q}[k':j]$ such that $i < k' \leq n$, and the three paths are also row merge fill paths in $H^{\times}(A)$. Our pivoting strategy interchanges rows i and k of A at the ith step of elimination. We use Algorithm 3 to decompose the path $\mathcal{Q}[k':j]$ in an alternating sequence of edges and middle paths. This sequence is transformed into a path from i to j in the graph $G(K)$ which has all of the intermediate column vertices smaller than i. This corresponds to an edge in the filled graph $G^{+}(K)$. Thus the element U_{ij} corresponds to a symbolic nonzero, and this ends our proof. ∎

We make one note about the structure prediction of $A = P_1L_1 \ldots P_{n-1}L_{n-1}\, U$. The tight bound of U obtained for the structure prediction of $PA = LU$ (Theorem 4.3) also represents a tight bound for U obtained in $A = P_1L_1 \ldots P_{n-1}L_{n-1}\, U$. But there does not seem to be a simple way to express tight exact bounds for $\widetilde{L}$, where $\widetilde{L}$ is the union of the L_i obtained from $A = P_1L_1 \ldots P_{n-1}L_{n-1}U$.

6. Concluding remarks. In this paper we have discussed two aspects of interest in the structure prediction problem of sparse LU factorization with partial pivoting of a matrix A. The first aspect considers the computation of the nonzero structure of the factors during Gaussian elimination with row interchanges. We have presented new results that provide an exact structure prediction for matrices that satisfy the strong Hall property or only the Hall property. We then have used the theoretical results to derive an algorithm for computing fill-ins. The second aspect is to estimate tight bounds of the structure of L and U prior to the numerical factorization. We have introduced tight exact bounds for the nonzero structure of L and U of Gaussian elimination with partial pivoting $PA = LU$, under the assumption that the matrix A satisfies the Hall property. We have also shown that the row merge graph represents a tight symbolic bound for the structure of the factors $\widetilde{L}$ and U obtained from the factorization $A = P_1L_1 \ldots P_{n-1}L_{n-1}U$.

The practical usage of the exact structure prediction presented in this paper remains an open problem. Several aspects are of interest. One important question is to understand if rounding to zero elements that correspond to numeric cancellation in exact arithmetic leads to instability in the Gaussian elimination. A different aspect is to analyze on real world matrices how many numeric cancellations, that Theorem 3.4 identifies, occur during Gaussian elimination. Another aspect is to compare experimentally the bounds presented in this paper with the bounds provided by the row merge graph, knowing that the latter can be efficiently computed [14].

Acknowledgments. The authors thank the anonymous reviewers for their helpful comments and suggestions to improve the presentation of the paper.

REFERENCES

[1] R. K. Brayton, F. G. Gustavson, and R. A. Willoughby, *Some results on sparse matrices*, Math. Comp., 24 (1970), pp. 937–954.

[2] R. A. Brualdi and H. J. Ryser, *Combinatorial Matrix Theory*, Cambridge University Press, Cambridge, 1991.

[3] R. A. Brualdi and B. L. Shader, *Strong Hall matrices*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 359–365.

[4] T. F. Coleman, A. Edenbrandt, and J. R. Gilbert, *Predicting fill for sparse orthogonal factorization*, J. ACM, 33 (1986), pp. 517–532.

[5] J. W. Demmel, J. R. Gilbert, and X. S. Li, *An asynchronous parallel supernodal algorithm for sparse Gaussian elimination*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 915–952.

[6] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 720–755.

[7] I. S. Duff, *On algorithms for obtaining a maximum transversal*, ACM Trans. Math. Software, 7 (1981), pp. 315–330.

[8] F. R. Gantmacher, *The Theory of Matrices*, Chelsea, New York, 1959.

[9] A. George and E. Ng, *Symbolic factorization for sparse Gaussian elimination with partial pivoting*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 877–898.

[10] J. R. Gilbert and J. W. H. Liu, *Elimination structures for unsymmetric sparse LU factors*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 334–352.

[11] J. R. Gilbert and E. G. Ng, *Predicting structure in nonsymmetric sparse matrix factorizations*, in Graph Theory and Sparse Matrix Computation, A. George, J. R. Gilbert, and J. W. H. Liu, eds., Springer-Verlag, New York, 1994, pp. 107–139.

[12] J. R. Gilbert, *An Efficient Parallel Sparse Partial Pivoting Algorithm*, Technical report 88/45052-1, Christian Michelsen Institute, Bergen, Norway, 1988.
[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 2nd ed., Baltimore, MD, 1989.
[14] L. Grigori, M. Cosnard, and E. G. Ng, *On the row merge tree for sparse LU factorization with partial pivoting*, BIT, 47 (2007), pp. 45–76.
[15] D. R. Hare, C. R. Johnson, D. D. Olesky, and P. van den Driessche, *Sparsity analysis of the QR factorization*, SIAM J. Matrix Anal. Appl., 14 (1991), pp. 655–669.
[16] X. Li and J. Demmel, *A scalable distributed-memory sparse direct solver for unsymmetric linear systems*, ACM Trans. Math. Software, 29 (2003), pp. 110–140.
[17] L. Lovasz and M. D. Plummer, *Matching Theory*, North–Holland, Amsterdam, 1986.
[18] A. Pothen and C.-J. Fan, *Computing the block triangular form of a sparse matrix*, ACM Trans. Math. Software, 16 (1990), pp. 303–324.
[19] A. Pothen, *Predicting the structure of sparse orthogonal factors*, Linear Algebra Appl., 194 (1993), pp. 183–204.
[20] D. J. Rose and R. E. Tarjan, *Algorithmic aspects of vertex elimination on directed graphs*, SIAM J. Appl. Math., 34 (1978), pp. 176–197.
[21] K. Shen, T. Yang, and X. Jiao, *S+: Efficient 2d sparse LU factorization on parallel machines*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 282–305.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1546–1565

ERROR ESTIMATES FOR POLYNOMIAL KRYLOV APPROXIMATIONS TO MATRIX FUNCTIONS*

FASMA DIELE†, IGOR MORET‡, AND STEFANIA RAGNI§

Abstract. In this paper we are interested in the polynomial Krylov approximations for the computation of $\varphi(A)v$, where A is a square matrix, v represents a given vector, and φ is a suitable function which can be employed in modern integrators for differential problems. Our aim consists of proposing and analyzing innovative a posteriori error estimates which allow a good control of the approximation procedure. The effectiveness of the results we provide is tested on some numerical examples of interest.

Key words. Krylov methods, matrix functions, exponential-like functions, error estimates

AMS subject classifications. 65F10, 65L05

DOI. 10.1137/070688924

1. Introduction.

Krylov subspace methods are commonly used for evaluating $y = \varphi(A)v$ in the case when φ is a suitable function, A represents a large sparse matrix, and v is a given vector. The simplest Krylov approach provides polynomial approximations achieved in the Krylov subspaces generated by A and v. Its application to the exponential matrix operator and, more in general, to functions representing differential problem solutions has received wide attention in the literature; among others, we quote [7], [22], [31], [17], [26], [23], where different a priori error estimates have been proposed. We notice that these estimates are usually obtained by employing some minimal information concerning the matrix A, such as the spectrum size or the numerical range; therefore, they often turn out to be pessimistic and not so useful in the error-control procedure due to their inability of adaption to the spectrum. Thus, obtaining adaptive a posteriori error bounds represents an interesting task which, in our knowledge, has not been developed yet.

In this framework our aim consists of proposing a posteriori error estimates employed in the computation of a specific class of functions represented in integral form. In particular, we refer to exponential-like operators that are considered as the basis for constructing exponential integrators widely used in modern methods for solving differential problems (see, e.g., [19], [20], [18], [21], [25], [2]); moreover, we also analyze the case of trigonometric functions which are involved in the solution of second order differential systems (see [15]).

In this respect, the paper is organized in the following way. In section 2 we provide some general results related to polynomial Krylov approximations for matrix functions represented by some general integral forms. We specify and extend the results dealing with the cases of both the exponential-like functions and the trigonometric ones, in the specific of *cos* and *sinc* functions, in section 3. Moreover, some numer-

*Received by the editors April 23, 2007; accepted for publication (in revised form) by V. Simoncini July 22, 2008; published electronically December 19, 2008.
http://www.siam.org/journals/simax/30-4/68892.html

†Istituto per le Applicazioni del Calcolo M. Picone, CNR, Via Amendola 122, 70126 Bari, Italy (f.diele@ba.iac.cnr.it).

‡Dipartimento di Matematica e Informatica, Università di Trieste, Via Valerio 12/1, 34127 Trieste, Italy (moret@units.it).

§Corresponding author. Facoltà di Economia, Università di Bari, Via Camillo Rosalba 53, 70124 Bari, Italy (s.ragni@ba.iac.cnr.it).

ical experiments are shown in section 4 with the aim of testing the effectiveness of the proposed estimates. Our bounds are compared with the best a priori error estimates already known in the literature (see [7] and [17]) since, up to our knowledge, a posteriori ones have never been provided so far.

2. Polynomial Krylov methods. We are going to sketch important and well-known features of polynomial Krylov methods; in particular, we provide and prove some results concerning their application in computing specific matrix functions represented in integral form. With this aim, in the sequel $\|\cdot\|$ represents the vector and matrix Euclidean norm and Π_k is the set of all the algebraic polynomials with degree equal or less than k ($k \in \mathbb{N}$).

Let A be a given $N \times N$ matrix; we denote the spectrum by $\sigma(A)$ and the *numerical range* by $W(A)$, i.e.,

$$W(A) = \left\{ \frac{\langle x, Ax \rangle}{\langle x, x \rangle},\ x(\neq 0) \in \mathbb{C}^N \right\},$$

where $\langle \cdot, \cdot \rangle$ represents the Euclidean inner product. We consider a given vector v with $\|v\| = 1$, and we focus on the Krylov subspaces

$$K_m(A, v) = span\left\{v, Av, \ldots, A^{m-1}v\right\}$$

related to A and v; thus, we suppose that $\{v_1, v_2, \ldots, v_j, \ldots\}$ is an ordered system of vectors providing a basis for each Krylov subspace, that is

$$K_m(A, v) = span\left\{v_1, v_2, \ldots, v_m\right\}$$

for every m. Moreover, under the assumption that

$$Av_j = \sum_{i=1}^{j+1} h_{i,j} v_i, \tag{1}$$

where $h_{i,j}$'s are suitable coefficients with $h_{j+1,j} \neq 0$ for $1 \leq j \leq m-1$, we denote by H_m the $m \times m$ upper Hessenberg matrix having its entries given by the values $h_{i,j}$ ($1 \leq i, j \leq m$). It follows that

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T, \tag{2}$$

where V_m=$(v_1, v_2, \ldots, v_m)$ and each e_j represents the jth column of the $m \times m$ unit matrix.

It is possible to prove that for any $q_{m-1} \in \Pi_{m-1}$ it holds that

$$q_{m-1}(A)v = V_m q_{m-1}(H_m) e_1. \tag{3}$$

Different choices for the previous basis $\{v_1, v_2, \ldots, v_m\}$, generated via relationships like (1), have been proposed in the literature: for instance, it is possible to adopt classical, full, or incomplete (IOM) Arnoldi's algorithm, Lanczos biorthogonalization, Chebyshev or Faber polynomials ([26]), quasi-kernel polynomials ([27]), or Leja sequences ([4]). Furthermore, throughout the paper we assume exact arithmetic (anyway, we refer to the recent survey [24] and the references therein concerning the treatment of problems related to finite precision arithmetic in the field of Krylov methods).

As already pointed out, our aim consists of evaluating

$$y = \varphi(A)v$$

under the assumption that φ is a function representable in the general integral form

$$\varphi(A) = \int_\Gamma g(\lambda)(\lambda I + A)^{-1} d\lambda, \tag{4}$$

where Γ and g are a suitable curve in the complex plane and a scalar function, respectively. We notice that every function analytic in a domain including $\sigma(A)$ can be written in this integral form due to the well-known Dunford–Taylor theorem. We remark also that a specific example in the class of functions (4) consists of the discrete counterpart given by the partial fraction forms. They may arise by applying any quadrature to evaluate integral representation (4), as it is shown in some applications in [3], [6], [21]. Moreover, linear partial fractions can give raise to some computational advantages when they are adopted for representing rational functions; in this respect, we quote the recent paper [23], where the convergence of the polynomial Krylov method is investigated as what concerns Padé or Chebyshev approximations of the exponential function.

Assume that $\sigma(A) \cup \sigma(H_m)$ is included in a given domain G; furthermore let φ be analytic in G itself and continuous on its closure. Then the mth polynomial approximation (related to (2)) for the computation of y is defined by

$$y_m = V_m \varphi(H_m) e_1.$$

We recall that, in particular, if $\{v_j\}_{j=1,2,\ldots}$ are generated by means of the full Arnoldi's algorithm, they are orthonormal vectors and the resulting approach is referred to as the Polynomial Arnoldi (PA) method. More in general, from (2) and the relationship

$$y - y_m = \int_\Gamma g(\lambda) \left((\lambda I + A)^{-1} v - V_m (\lambda I + H_m)^{-1} e_1 \right) d\lambda, \tag{5}$$

it follows that for any polynomial Krylov scheme we have

$$y - y_m = \int_\Gamma g(\lambda) \xi_m(\lambda) (\lambda I + A)^{-1} v_{m+1} d\lambda, \tag{6}$$

where

$$\xi_m(\lambda) = -h_{m+1,m} \left(e_m^T (\lambda I + H_m)^{-1} e_1 \right).$$

By exploiting (6) it is possible both to obtain error estimates and to put a restart procedure right; indeed the error formulation is featured by the same form of function φ (different approaches of restarting Krylov methods for matrix functions can be found in [5] and [8]).

We are going to propose another error representation which generalizes a result already given in [17] and [32] for the Arnoldi's method. With this aim we set

$$\overline{H}_{\lambda,m} = \begin{pmatrix} \lambda I + H_m \\ h_{m+1,m} e_m^T \end{pmatrix} \in \mathbb{C}^{(m+1)\times(m+1)};$$

thus from (2) we obtain

$$(\lambda I + A) V_m = V_{m+1} \overline{H}_{\lambda,m}. \tag{7}$$

In the sequel, $\overline{e}_1$ represents the first column of the $(m+1)\times(m+1)$ unit matrix and $\{U_j\}_{j=1,2,\ldots}$ is a sequence of $N\times j$ matrices which satisfy the relationships $U_j^H V_j = I$ and $U_j^H v_{j+1} = 0$; using these notations, in the following lemma (see [17] for its proof) a further error formula is given.

LEMMA 1. *For each $p_m \in \Pi_m$ with $p_m(-\lambda) = 1$, it holds that*

$$y - y_m = \int_\Gamma g(\lambda)\left((\lambda I + A)^{-1} - V_m(\lambda I + H_m)^{-1}U_m^H\right) p_m(A)v\, d\lambda.$$

Moreover, for every $x \in \mathbb{C}^m$ the error formula is given in the equivalent form

$$y - y_m = \int_\Gamma g(\lambda)\left((\lambda I + A)^{-1} - V_m(\lambda I + H_m)^{-1}U_m^H\right) V_{m+1}(\overline{e}_1 - \overline{H}_{\lambda,m}x)\, d\lambda.$$

In the sequel, for any $m \geq 0$, we will denote

$$\mu_m(\lambda, A) = \min_{p_m\in\Pi_m,\ p_m(-\lambda)=1} \|p_m(A)v\|$$

and

$$\nu_m(\lambda, A) = \min_{x\in\mathbb{C}^m} \left\|\overline{e_1} - \overline{H}_{\lambda,m}x\right\|.$$

Using (7) it is not so difficult to verify that

$$\left\|U_{m+1}^H\right\|^{-1} \nu_m(\lambda, A) \leq \mu_m(\lambda, A) \leq \|V_{m+1}\|\, \nu_m(\lambda, A).$$

In particular, when the PA method is applied ($U_m = V_m$), Lemma 1 yields

$$\|y - y_m\| \leq \int_\Gamma |g(\lambda)| \left(\left\|(\lambda I + A)^{-1}\right\| + \left\|(\lambda I + H_m)^{-1}\right\|\right) \mu_m(\lambda, A)\, |d\lambda|. \tag{8}$$

For our purposes we set $\delta_m = 1 + h_{m+1,m}^2 \|(\lambda I + H_m)^{-H} e_m\|^2$ and consider $\overline{H}_{\lambda,m}$ partitioned in the form

$$\overline{H}_{\lambda,m} = \begin{pmatrix} d_{\lambda,m}^H \\ T_{\lambda,m} \end{pmatrix}, \tag{9}$$

where $d_{\lambda,m}^H$ represents the first row of $\overline{H}_{\lambda,m}$ itself and $T_{\lambda,m}$ is an $m\times m$ upper triangular matrix with diagonal entries given by the $h_{j+1,j}$'s ($j = 1,\ldots,m$).

LEMMA 2. *For every $\lambda \notin \sigma(A)$ it holds that*

$$\nu_m(\lambda, A) = \frac{1}{\sqrt{1 + \left\|T_{\lambda,m}^{-H} d_{\lambda,m}\right\|^2}} \tag{10}$$

or, equivalently,

$$\nu_m(\lambda, A) = \delta_m^{-1/2}\, |\xi_m(\lambda)| \tag{11}$$

where

$$|\xi_m(\lambda)| = \left|\det(\lambda I + H_m)^{-1} \prod_{j=1}^m h_{j+1,j}\right|. \tag{12}$$

Moreover, for $m > 1$, an upper bound is given by

$$\nu_m(\lambda, A) \leq h_{m+1,m} \left\| U_m(\lambda I + H_m)^{-H} e_m \right\| \delta_m^{-1/2} \mu_{m-1}(\lambda, A). \tag{13}$$

Proof. Identity (10) can be proved using the same arguments as employed in [33], Proposition 4.1. Moreover it is possible to verify that

$$\langle e_m, p_{m-2}(H_m) e_1 \rangle = 0 \tag{14}$$

for every $p_{m-2} \in \Pi_{m-2}$. Therefore, exploiting the Cayley–Hamilton theorem, (12) follows. Furthermore, we point out that $\overline{H}_{\lambda,m}^H \overline{H}_{\lambda,m} = d_{\lambda,m} d_{\lambda,m}^H + T_{\lambda,m}^H T_{\lambda,m}$, and thus we obtain

$$\prod_{j=1}^{m} h_{j+1,j}^2 = \det\left(T_{\lambda,m}^H T_{\lambda,m}\right) = \det\left(\overline{H}_{\lambda,m}^H \overline{H}_{\lambda,m}\right) \left(1 + \left\| T_{\lambda,m}^{-H} d_{\lambda,m} \right\|^2\right)^{-1} \tag{15}$$

and $\det(\overline{H}_{\lambda,m}^H \overline{H}_{\lambda,m}) = \delta_m \det((\lambda I + H_m)^H (\lambda I + H_m))$. In this way, once (10) is proved, we have

$$\nu_m(\lambda, A) = \delta_m^{-1/2} \left| \prod_{j=1}^{m} h_{j+1,j} \det((\lambda I + H_m))^{-1} \right|,$$

and then (11) follows from (12). Furthermore, due to (14) we obtain

$$\nu_m(\lambda, A) = \delta_m^{-1/2} \left| h_{m+1,m} \langle e_m, ((\lambda I + H_m)^{-1} + p_{m-2}(H_m)) e_1 \rangle \right|$$

for every $p_{m-2} \in \Pi_{m-2}$ ($m \geq 2$). Thus, when we consider any $p_{m-1} \in \Pi_{m-1}$ with $p_{m-1}(-\lambda) = 1$, we have

$$\nu_m(\lambda, A) = \delta_m^{-1/2} \left| h_{m+1,m} \langle e_m, (\lambda I + H_m)^{-1} p_{m-1}(H_m) e_1 \rangle \right|.$$

Due to this inequality and to (3) it follows that (13) is proved. ∎

In the remaining part of the section we suppose that the full Arnoldi method is employed in the construction of the Krylov subspaces; of course, in this case, it holds that $\mu_m(\lambda, A) = \nu_m(\lambda, A)$. Moreover, the results in Lemma 2 point out the $h_{j+1,j}$'s role in the Krylov procedure. It is well known that $h_{j+1,j} \geq 0$ for each j and the following relationship can be proved (see [34, p. 269])

$$\prod_{j=1}^{m} h_{j+1,j} = \min_{q_m \in \Pi_m^{(0)}} \|q_m(A) v\| \tag{16}$$

with $\Pi_m^{(0)}$ being the set of all the monic polynomials with degree m. Recall that (16) is useful for developing a priori bounds for $\prod_{j=1}^{m} h_{j+1,j}$ by means of known classical results. For instance, if A is self-adjoint with spectrum in an interval of length 4γ, then by suitable bounds for Chebyshev polynomials it is possible to verify that (see [13, p. 91])

$$\prod_{j=1}^{m} h_{j+1,j} \leq 2\gamma^m. \tag{17}$$

A further upper bound can be obtained assuming that the distinct eigenvalues of A are arranged in decreasing order

$$0 \leq \cdots < \lambda_j(A) < \lambda_{j-1}(A) < \cdots < \lambda_2(A) < \lambda_1(A)$$

and choosing $p_m(\lambda) = \prod_{j=1}^{m}(\lambda - \lambda_j(A))$ in (16); indeed, under this assumption we have

$$\prod_{j=1}^{m} h_{j+1,j} \leq \max_{k>m} \prod_{j=1}^{m} (\lambda_j(A) - \lambda_k(A)). \tag{18}$$

More in general, for any scalar α we consider the singular values of $\alpha I + A$ given by

$$\sigma_1(\alpha I + A) \geq \sigma_2(\alpha I + A) \geq \cdots \geq \sigma_j(\alpha I + A) \geq \cdots$$

Since $V_m V_m^H$ is an orthogonal projection, from (15) and (10) it is possible to obtain (see [14, p. 128])

$$\begin{aligned} \Pi_{j=1}^{m} h_{j+1,j} &= \mu_m(\alpha, A) \sqrt{\det\left(V_m^H \left(\alpha I + A^H\right)(\alpha I + A) V_m\right)} \\ &\leq \mu_m(\alpha, A) \prod_{j=1}^{m} \sigma_j(\alpha I + A). \end{aligned}$$

Assuming $\alpha I + A$ to be a compact operator in a Hilbert space with $q > 0$ summable singular values (i.e., $\sum_{j=1}^{+\infty} \sigma_j^q < +\infty$), then by the geometric-arithmetic mean inequality we have

$$\prod_{j=1}^{m} \sigma_j(\alpha I + A) \leq \left(\frac{1}{m} \sum_{j=1}^{m} (\sigma_j(\alpha I + A))^q \right)^{m/q}. \tag{19}$$

In particular, for Hilbert–Schmidt operators, in correspondence with $q = 2$, it holds that

$$\begin{aligned} \prod_{j=1}^{m} \sigma_j(\alpha I + A) &\leq \left(\frac{1}{m} \sum_{j=1}^{m} (\sigma_j(\alpha I + A))^2 \right)^{m/2} \\ &\leq \left(\frac{1}{m} \|\alpha I + A\|_F^2 \right)^{m/2} \end{aligned}$$

with $\|\cdot\|_F$ being the Frobenius norm.

3. Application to specific functions. We are interested in applying the results we have just provided in the previous section to specific cases. More precisely, we first consider the so-called *φ-functions* which are widely used in modern methods for solving differential problems; then we account for trigonometric functions, like *cos* and *sinc* ones, that are involved in the construction of several numerical integrators for solving second-order differential systems. We consider the PA method (anyway, the use of other bases can be taken into account by developing similar arguments).

3.1. The case of exponential-like functions. It is well known that *φ-functions* are exponential-like ones defined as

$$\varphi_0(-tz) = \exp(-tz),$$
$$\varphi_k(-tz) = \frac{1}{(k-1)!t^k}\int_0^t \exp(-(t-s)z)s^{k-1}ds, \qquad k = 1, 2, \ldots,$$

and they satisfy the recursive formula

$$\varphi_{k+1}(u) = \frac{\varphi_k(u) - \frac{1}{k!}}{u}, \quad \varphi_k(0) = \frac{1}{k!}, \qquad k = 0, 1, 2, \ldots \tag{20}$$

With the aim of evaluating

$$y(k;t) = \varphi_k(-tA)v, \qquad t > 0, \tag{21}$$

we recall that $W(H_m) \subseteq W(A)$ holds for each m and we denote by

$$y_m(k;t) = V_m\varphi_k(-tH_m)e_1$$

the mth approximation to $y(k;t)$. We would like to remark that, as pointed out by other authors (see, e.g., [21], [25], [2]), the computation of these functions may be affected by some difficulties. Concerning the different approaches carried out in the literature, we quote the one related to Padè approximants discussed in [2].

For $a \geq 0$ and $0 \leq \vartheta \leq \pi/2$ let us define the set

$$\Sigma_{\vartheta,a} = \{\lambda \in \mathbb{C} : |\arg(\lambda - a)| \leq \vartheta\}$$

and assume that

$$W(A) \subset \Sigma_{\vartheta,a}. \tag{22}$$

It is well known that this assumption holds in important applications such as parabolic partial differential equations.

As we already mentioned, our aim consists of introducing novel error estimates, which will be compared with other results already provided in the literature. In this respect we remark that a priori error bounds can be found in [7], [22], [31], [17], [26], [23]. For instance, we refer to the following results given in [17].

PROPOSITION 3. *Under the assumption that A is Hermitian with eigenvalues in the interval $[0, 4\gamma]$, an error estimate for the PA approximation is given by*

$$\|y(0;t) - y_m(0;t)\| \leq \left(12\frac{\gamma t}{m^2} + 8\frac{\sqrt{\gamma t}}{m}\right)\exp\left(\frac{-\beta m^2}{4\gamma t}\right), \tag{23}$$

for $\sqrt{4\gamma t} \leq m \leq 2\gamma t$, where $\beta > 0.92$ is a suitable parameter, and

$$\|y(0;t) - y_m(0;t)\| \leq \left(\frac{5}{\gamma t} + 3\sqrt{\frac{\pi}{\gamma t}}\right)\exp\left(\frac{(\gamma t)^2}{m}\right)\exp(-2\gamma t)\left(\frac{\exp(1)\gamma t}{m}\right)^m, \tag{24}$$

for $m > 2\gamma t$.

Moreover, assuming A is skew-Hermitian with eigenvalues in an interval of the imaginary axis with length 4γ, then the PA approximation error is bounded by

$$\|y(0;t) - y_m(0;t)\| \leq \frac{1}{3}\left(\frac{4}{\gamma t} + \frac{11}{\sqrt{\gamma t}}\right)\exp\left(\frac{-(\gamma t)^2}{m}\right)\left(\frac{\exp(1)\gamma t}{m}\right)^m, \tag{25}$$

for $m \geq 2\gamma t (\gamma t > \frac{1}{2})$.

Moreover, we take into account also the following a priori error bounds which can be obtained by the argument developed in [7]. In several cases, these estimates reveal sharper than the previous ones as we show in the numerical tests.

PROPOSITION 4. *Let A be symmetric with eigenvalues in the interval $[0, 4\gamma]$; then, an upper bound for the error in the PA approximation is obtained as*

$$\|y(0;t) - y_m(0;t)\| \leq 4\exp(-2t\gamma)\sum_{k=m}^{+\infty} I_k(2t\gamma), \tag{26}$$

where I_k represents the modified Bessel function of the first kind. Moreover, under the assumption that A is skew-Hermitian with eigenvalues in an interval of the imaginary axis with length 4γ, then the PA approximation error is bounded by

$$\|y(0;t) - y_m(0;t)\| \leq 4\sum_{k=m}^{+\infty} |J_k(2t\gamma)|, \tag{27}$$

where J_k is the Bessel function of the first kind.

Our analysis is based on the integral representation (cfr. [9, p. 234] and [29, pp. 28–29])

$$y(k;t) = \frac{1}{t^k}\lim_{n\to\infty}\frac{1}{2\pi i}\int_{\epsilon-in}^{\epsilon+in}\exp(t\lambda)\lambda^{-k}(\lambda I + A)^{-1}v d\lambda, \quad k = 0, 1, 2, \ldots,$$

which, under our assumptions, holds for every $\varepsilon > 0$ with uniform convergence when t is chosen in compact intervals of $(0, +\infty)$. More precisely, in [9] and [29], this representation is proved for $k = 0, 1, 2$, but it is possible to verify that it can be extended also for $k > 2$ by noting that

$$\lim_{n\to\infty}\frac{1}{2\pi i}\int_{\epsilon-in}^{\epsilon+in}\lambda^{-k}(\lambda I + A)^{-1}v d\lambda = 0.$$

Given $\varepsilon > 0$, we set $\lambda = \varepsilon + i\rho$, and then we obtain

$$y(k;t) = \frac{1}{t^k}\exp(t\varepsilon)\lim_{n\to\infty}\frac{1}{2\pi}\int_{-n}^{+n}\exp(it\rho)(\varepsilon + i\rho)^{-k}((\varepsilon + i\rho)I + A)^{-1}v d\rho. \tag{28}$$

In the sequel we suppose A and v to be real, and then we consider $\mu_j = a_j + ib_j$ $(j = 1, 2, \ldots, m)$ the eigenvalues of matrix H_m arranging them as $\mu_1, \ldots, \mu_{m_1}$ the real ones and $\mu_{m_1+1}, \ldots, \mu_m$ the complex conjugate ones. Moreover we set

$$r_j = |\mu_j| = \left((\varepsilon + a_j)^2 + b_j^2\right)^{1/2} \quad \text{for } j = 1, \ldots, m,$$

$$R = \max_{1\leq j\leq m} r_j$$

and

$$\omega_m(\varepsilon) = \prod_{j=1}^{m}(r_j(\varepsilon + a_j))^{1/2}. \tag{29}$$

For convenience we also define the functions

$$\beta_m(\rho) = \prod_{j=1}^{m_1}\left(1 + \rho^2/r_j^2\right)^{1/2}\prod_{j=m_1+1}^{m}\left(1 + \rho^2/r_j^2\right)^{1/4} \tag{30}$$

and

$$d_m(\rho) = \frac{\prod_{j=1}^{m} h_{j+1,j}}{\omega_m(\varepsilon)\beta_m(\rho)}.$$

Under these assumptions, applying (8) we will prove the following result.

PROPOSITION 5. *Consider an arbitrary value $\varepsilon > 0$ and suppose $m + m_1 + 2k \geq 4$. Then, for the PA method the following error bound holds:*

$$\|y(k;t) - y_m(k;t)\| \leq c_{k,m} \int_0^\infty \frac{\left(1 + \rho^2/\varepsilon^2\right)^{-k/2}}{\beta_m(\rho)\sqrt{1 + d_m(\rho)^2}} d\rho \tag{31}$$

where

$$c_{k,m} = \frac{2\exp(t\varepsilon)\prod_{j=1}^{m} h_{j+1,j}}{\pi\omega_m(\varepsilon)(\varepsilon + a)(t\varepsilon)^k}. \tag{32}$$

Proof. For $\rho \geq 0$ we have

$$\begin{aligned} |\det((\varepsilon + i\rho)I + H_m)| &= \prod_{j=1}^{m} |(\varepsilon + i\rho) + \mu_j| \\ &= \prod_{j=1}^{m} \left({r_j}^2 + \rho^2 + 2\rho b_j\right)^{1/2}. \end{aligned}$$

We claim that

$$|\det((\varepsilon + i\rho)I + H_m)| \geq \omega_m(\varepsilon)\beta_m(\rho). \tag{33}$$

Indeed, when $m_1 = m$ then the previous relationship clearly holds; otherwise, since H_m is real, there are $(m - m_1)/2$ couples of conjugate eigenvalues. For each of them, say $\mu^{\pm} = \overline{a} \pm i\overline{b}$ ($\overline{b} > 0$), we have

$$\begin{aligned} |(\varepsilon + i\rho) + \mu^+|\,|(\varepsilon + i\rho) + \mu^-| &= \left(\left((\varepsilon + \overline{a})^2 + (\rho + \overline{b})^2\right)\left((\varepsilon + \overline{a})^2 + (\rho - \overline{b})^2\right)\right)^{1/2} \\ &\geq \left((\varepsilon + \overline{a})^2 + \overline{b}^2 + \rho^2\right)^{1/2} (\varepsilon + \overline{a}); \end{aligned}$$

hence (33) follows. Using the same arguments, we obtain (33) also for $\rho < 0$. Therefore, due to (12) we have

$$|\xi_m(\varepsilon + i\rho)| \leq d_m(\rho). \tag{34}$$

Moreover we account for the following bound

$$\max\left(\left\|((\varepsilon + i\rho)I + A)^{-1}\right\|, \left\|((\varepsilon + i\rho)I + H_m)^{-1}\right\|\right) \leq (\varepsilon + a)^{-1}, \tag{35}$$

which arises from the well-known inequality $\|(\lambda I + A)^{-1}\| \leq dist(-\lambda, W(A))^{-1}$ and by assumption (22). Then, referring to representation (28), we consider error formula (8). From (11) and (13) ($U_m = V_m$) we obtain

$$|\xi_m(\lambda)| \leq h_{m+1,m} \left\|(\lambda I + H_m)^{-H} e_m\right\| \mu_{m-1}(\lambda, A).$$

Thus, due to $\mu_{m-1}(\lambda, A) \leq \|v\| = 1$, from (11) it follows that

$$\mu_m(\lambda, A) \leq \frac{|\xi_m(\lambda)|}{\sqrt{1 + |\xi_m(\lambda)|^2}}. \tag{36}$$

As a consequence, by means of (34) and (35), we have

$$\|y(k;t) - y_m(k;t)\| \leq \frac{2\exp(t\varepsilon)}{\pi(\varepsilon + a)t^k} \int_0^\infty \frac{d_m(\rho)d\rho}{(\varepsilon^2 + \rho^2)^{k/2}\sqrt{1 + d_m(\rho)^2}}.$$

Thus (31) is proved. We notice that if $m + m_1 + 2k \geq 4$, then the integral in (31) converges. □

We would like to point out that by (31) we can recognize the well-known superlinear convergence of the method; indeed, setting $\varepsilon = m/t$, for a suitable constant C, we obtain

$$\|y(k;t) - y_m(k;t)\| \leq \frac{Ct}{m^{k+1}} \left(\frac{t\exp(1)(\prod_{j=1}^m h_{j+1,j})^{1/m}}{m} \right)^m,$$

which stresses the dependence on the term $t(\prod_{j=1}^m h_{j+1,j})^{1/m}$. A further speed-factor may be added and a consequent *super-super linear* convergence is reached when the product $\prod_{j=1}^m h_{j+1,j}$ rapidly decreases; but, unfortunately, this does not occur in the case of approximating parabolic problem solution, where A represents the discretization of an elliptic operator and its spectrum enlarges as the mesh is refined. Anyway, in order to overcome this drawback, we mention the use of a rational Krylov method recently adopted in [28] and [10] (numerical comparisons can be found also in the recent paper [30]). Moreover, though a priori information on the behavior of $\prod_{j=1}^m h_{j+1,j}$ can be provided by (17), (18), (19), and (2), in our experiments this product is directly evaluated in the PA process.

The result we give below aims to avoid the use of a quadrature rule for evaluating the integral in (31).

PROPOSITION 6. *Let $\varepsilon > 0$ and assume $m + m_1 \geq 4$. Referring to the previous notations, the approximation error obtained by applying the PA method is bounded by*

$$\|y(k;t) - y_m(k;t)\| \leq c_{k,m} C_m \tag{37}$$

where we set

$$C_m = \frac{\sqrt{\pi}}{2\sqrt{(1 + d_m(\varepsilon)^2)\, S_m^{(1)}}} + \frac{\exp\left(-\varepsilon^2 S_m^{(2)}\right)}{\sqrt{1 + d_m(R)^2}}(R - \varepsilon) + \left(\frac{\varepsilon}{\sqrt{\varepsilon^2 + R^2}}\right)^k \frac{R\pi}{2^{\left(\frac{m+m_1}{4}+1\right)}}$$

with

$$S_m^{(1)} = \frac{k}{4\varepsilon^2} + \frac{1}{2}\sum_{i=1}^{m_1} \frac{1}{r_j^2 + \varepsilon^2} + \frac{1}{4}\sum_{i=m_1+1}^{m} \frac{1}{r_j^2 + \varepsilon^2}$$

and

$$S_m^{(2)} = \frac{k}{2(\varepsilon^2 + R^2)} + \frac{1}{2}\sum_{i=1}^{m_1} \frac{1}{r_j^2 + R^2} + \frac{1}{4}\sum_{i=m_1+1}^{m} \frac{1}{r_j^2 + R^2}.$$

Proof. Setting

$$\Psi_m(\rho) = \frac{(1+\rho^2/\varepsilon^2)^{-k/2}}{\beta_m(\rho)},$$

we approximate the integral in (31), exploiting the relationship

$$\Psi_m(\rho) \le \exp\left(-\rho^2\left(\frac{k}{2(\varepsilon^2+\rho^2)} + \frac{1}{2}\sum_{i=1}^{m}\frac{1}{r_j^2+\rho^2} + \frac{1}{4}\sum_{i=m_1+1}^{m}\frac{1}{r_j^2+\rho^2}\right)\right), \tag{38}$$

which comes from $1-x \le \exp(-x)$, $0 \le x \le 1$. At first we have

$$\int_0^\varepsilon \frac{\Psi_m(\rho)}{\sqrt{1+d_m(\rho)^2}}d\rho \le \frac{1}{\sqrt{1+d_m(\varepsilon)^2}}\int_0^\varepsilon \Psi_m(\rho)d\rho.$$

Thanks to (38), using formulae 7.1.1 and 7.1.2 in [1], we get

$$\begin{aligned}\int_0^\varepsilon \Psi_m(\rho)d\rho &\le \int_0^\varepsilon \exp\left(-\rho^2 S_m^{(1)}\right)d\rho = \frac{1}{\sqrt{S_m^{(1)}}}\int_0^{\varepsilon\sqrt{S_m^{(1)}}} \exp\left(-x^2\right)dx \\ &\le \frac{\sqrt{\pi}}{2\sqrt{S_m^{(1)}}}.\end{aligned}$$

Furthermore (38) yields

$$\int_\varepsilon^R \frac{\Psi_m(\rho)}{\sqrt{1+d_m(\rho)^2}}d\rho \le \frac{\int_\epsilon^R \exp\left(-\rho^2 S_m^{(2)}\right)d\rho}{\sqrt{1+d_m(R)^2}} \le \frac{\exp\left(-\varepsilon^2 S_m^{(2)}\right)(R-\varepsilon)}{\sqrt{1+d_m(R)^2}}.$$

Finally it is not so difficult to verify that, for $m+m_1 \ge 4$, it holds

$$\begin{aligned}\int_R^{+\infty} \frac{\Psi_m(\rho)}{\sqrt{1+d_m(\rho)^2}}d\rho &\le \int_R^{+\infty}\Psi_m(\rho)d\rho \\ &\le \left(\frac{\varepsilon}{\sqrt{\varepsilon^2+R^2}}\right)^k \int_R^{+\infty}\frac{1}{(1+\rho^2/R^2)^{\frac{m+m_1}{4}}}d\rho \\ &\le \left(\frac{\varepsilon}{\sqrt{\varepsilon^2+R^2}}\right)^k \frac{\pi R}{2^{\left(\frac{m+m_1}{4}+1\right)}}.\end{aligned}$$

This completes the proof. □

Concerning the choice of ε, of course a suitable value is the one minimizing the bound. In practice ε can be chosen by trying to reach the minimum value of a significant part of it: for instance, we can find $\varepsilon > 0$ which minimizes

$$\frac{\exp(t\varepsilon)}{(\varepsilon+a)\omega_m(\varepsilon)\varepsilon^k} \tag{39}$$

or, when the b_j's are negligible with respect to the a_j's, simply

$$\frac{\exp(t\varepsilon)}{(\varepsilon+a)\prod_{j=1}^m(\varepsilon+a_j)\varepsilon^k}. \tag{40}$$

Remark 7. After some computations, it is possible to prove that the following inequality holds:

$$\max\left(\left\|((\varepsilon+i\rho)I+A)^{-1}\right\|,\left\|((\varepsilon+i\rho)I+H_m)^{-1}\right\|\right)\leq\kappa(\varepsilon,\rho,\vartheta)^{-1}$$

where

$$\kappa(\varepsilon,\rho,\vartheta)=\sqrt{(\varepsilon+a)^2+\rho^2},\ \text{if}\ |\rho|\tan\theta\leq\varepsilon+a$$

and

$$\kappa(\varepsilon,\rho,\vartheta)=\frac{(\varepsilon+a)\tan\theta+|\rho|}{\sqrt{1+\tan^2\vartheta}},\ \text{if}\ |\rho|\tan\theta>\varepsilon+a.$$

This bound could be employed instead of (35); yet, in practice its use does not influence substantially the error estimates.

At the end of this subsection we account for the computation of

$$y^+(k;t)=\varphi_k(-itA)v,\quad t>0$$

under the assumption that A is symmetric and positive semidefinite; as it is well known this problem is related to the solution of the Schrödinger equation (see, for instance, [16]). In this respect, we refer to (2) and consider the mth PA-approximation

$$y_m^+(k;t)=V_m\varphi_k(-itH_m)e_1. \tag{41}$$

Denoting by μ_j $(j=1,2,\dots m)$ the eigenvalues of the Hermitian and positive semidefinite matrix H_m, we set

$$r_j^2=\left(\varepsilon^2+\mu_j^2\right)\quad\text{for } 1\leq j\leq m,\quad R=\max_{1\leq j\leq m}r_j,$$
$$\omega_m^+(\varepsilon)=\prod_{j=1}^m r_j\quad\text{and}\quad G_m=\prod_{j=1}^m(1-\mu_j/r_j)^{1/2}.$$

Furthermore we define

$$\beta_m^+(\rho)=\prod_{j=1}^m\left(1+\rho^2/{r_j}^2\right)^{1/2}\quad\text{and}\quad d_m^+(\rho)=\frac{\prod_{j=1}^m h_{j+1,j}}{\omega_m^+(\varepsilon)\beta_m^+(\rho)}.$$

Under these notations we will prove the following result.

PROPOSITION 8. *Let us consider an arbitrary $\varepsilon>0$. Supposing $m+k\geq 2$, the PA approximation* (41) *yields the error estimate*

$$\left\|y^+(k;t)-y_m^+(k;t)\right\|\leq c_{k,m}C_m \tag{42}$$

where we set

$$c_{k,m}=\frac{\exp(t\varepsilon)\prod_{j=1}^m h_{j+1,j}}{\pi\varepsilon\omega_m^+(\varepsilon)(t\varepsilon)^k}\left(1+\frac{1}{G_m}\right),$$

$$C_m=\frac{\sqrt{\pi}}{2\sqrt{\left(1+d_m^+(\varepsilon)^2\right)S_m^{(1)}}}+\frac{\exp\left(-\varepsilon^2S_m^{(2)}\right)}{\sqrt{1+d_m^+(R)^2}}(R-\varepsilon)+\left(\frac{\varepsilon}{\sqrt{\varepsilon^2+R^2}}\right)^k\frac{R\pi}{2^{\left(\frac{m}{2}+1\right)}}$$

with

$$S_m^{(1)} = \frac{k}{4\varepsilon^2} + \frac{1}{2}\sum_{j=1}^{m} \frac{1}{r_j^2 + \varepsilon^2} \quad \textit{and} \quad S_m^{(2)} = \frac{k}{2(\varepsilon^2 + R^2)} + \frac{1}{2}\sum_{j=1}^{m} \frac{1}{r_j^2 + R^2}.$$

Proof. It is similar to the proof of Propositions 5 and 6 with slight modifications due to the different structure of the spectra of matrices iH_m and H_m. We replace A by iA in (28) and H_m by iH_m in (12); thus, we obtain

$$|\xi_m(\varepsilon + i\rho)| = \frac{\prod_{j=1}^{m} h_{j+1,j}}{|\det((\varepsilon + i\rho)I + iH_m)|}.$$

Then, for $\rho \geq 0$ we have

$$\begin{aligned} |\det((\varepsilon + i\rho)I + iH_m)| &= \prod_{j=1}^{m} \left(\varepsilon^2 + (\rho + \mu_j)^2\right)^{1/2} = \prod_{j=1}^{m} \left({r_j}^2 + \rho^2 + 2\rho\mu_j\right)^{1/2} \\ &\geq \omega_m^+(\varepsilon)\beta_m^+(\rho), \end{aligned}$$

while for $\rho < 0$ we get

$$\begin{aligned} |\det((\varepsilon + i\rho)I + iH_m)| &= \prod_{j=1}^{m} \left(r_j^2 + \rho^2\right)^{1/2} \left(1 - \frac{2\mu_j\,|\rho|}{r_j^2 + \rho^2}\right)^{1/2} \\ &\geq \prod_{j=1}^{m} \left(r_j^2 + \rho^2\right)^{1/2} \left(1 - \frac{\mu_j}{r_j}\right)^{1/2} = \omega_m^+(\varepsilon)\beta_m^+(\rho)G_m. \end{aligned}$$

Hence, for $\rho > 0$ it holds $|\xi_m(\varepsilon + i\rho)| \leq d_m^+(\rho)$ and $|\xi_m(\varepsilon - i\rho)| \leq d_m^+(\rho)/G_m$. Therefore, due to (35) which holds for $a = 0$, arguing as in Proposition 5, from (6) and (36) by easy computations it follows that

$$\left\|y^+(k;t) - y_m^+(k;t)\right\| \leq c_{k,m} \int_0^\infty \frac{\left(1 + \rho^2/\varepsilon^2\right)^{-k/2}}{\beta_m^+(\rho)\sqrt{1 + d_m^+(\rho)^2}} d\rho.$$

By the same arguments as in the proof of Proposition 6, we obtain the bound

$$\int_0^{+\infty} \frac{\left(1 + \rho^2/\varepsilon^2\right)^{-k/2}}{\beta_m^+(\rho)\sqrt{1 + d_m^+(\rho)^2}} d\rho \leq C_m$$

and we prove the whole result. ∎

3.2. The case of trigonometric functions. Our interest is now focused on $\varphi(x) = \cos(t\sqrt{x})$ and the so-called *sinc* function defined as $\varphi(x) = \sin(t\sqrt{x})/\sqrt{x}$ which arise in the solution of second-order problems (see [15], [12], [7]). In particular, concerning the solution of hyperbolic equations, we account for the computation of

$$y^{(c)}(t) = \cos\left(t\sqrt{A}\right) v \quad t > 0 \tag{43}$$

under the assumption that A is real symmetric and positive semidefinite. In this respect, [7] provides the following error estimate.

PROPOSITION 9. *Let the eigenvalues of matrix A lie in the interval $[0, 4\gamma]$; then, an upper bound for the error in the PA approximation*

$$y_m^{(c)}(t) = V_m \cos\left(t\sqrt{H_m}\right) e_1$$

is given by

$$\left\| y^{(c)}(t) - y_m^{(c)}(t) \right\| \leq 4 \sum_{k=m}^{+\infty} |J_{2k}(2t\sqrt{\gamma})|, \tag{44}$$

where each J_k represents the Bessel function of the first kind.

In the sequel, we add the assumption that A is positive definite with spectrum in the interval $[a, +\infty)$ $(a > 0)$. Then, according to [11], we recall the representation

$$y^{(c)}(t) = \cos\left(t\sqrt{A}\right) v = \lim_{n\to\infty} \frac{1}{2\pi i} \int_{\epsilon-in}^{\epsilon+in} \exp(t\lambda)\lambda \left(\lambda^2 I + A\right)^{-1} v d\lambda$$

holds for every $\varepsilon > 0$ and $t \in (0, +\infty)$. In this way we have

$$y^{(c)}(t) = \exp(t\varepsilon) \lim_{n\to\infty} \frac{1}{2\pi} \int_{-n}^{n} \exp(it\rho)(\varepsilon + i\rho) \left((\varepsilon + i\rho)^2 I + A\right)^{-1} v \, d\rho, \tag{45}$$

and we provide the following result concerning the PA approximation.

PROPOSITION 10. *Suppose $m \geq 2$ and set*

$$R = \max_{1\leq j\leq m} \left(\varepsilon^2 + \mu_j\right),$$

where μ_j's are the eigenvalues of matrix H_m. Then, the error estimate

$$\left\| y^{(c)}(t) - y_m^{(c)}(t) \right\| \leq \frac{\exp(t\varepsilon)R}{2\sqrt{2}\varepsilon^{m+2}} \prod_{j=1}^{m} \frac{h_{j+1,j}}{(\varepsilon^2 + \mu_j)^{1/2}} \tag{46}$$

holds for each $\varepsilon > 0$ and $t > 0$.

Proof. We use the error representation (6). Thanks to the following relationship

$$\left\| \left((\varepsilon + i\rho)^2 I + A\right)^{-1} \right\| = \min_{x\in\sigma(A)} \left| \left((\varepsilon^2 - \rho^2) + x\right) + 2i\varepsilon\rho)^{-1} \right|$$

it is not so difficult to verify that

$$\left\| \left((\varepsilon + i\rho)^2 I + A\right)^{-1} \right\| \leq \left(\left(\varepsilon^2 + \rho^2\right)^2 + a^2 + 2a\left(\varepsilon^2 - \rho^2\right)\right)^{-1/2}, \text{ if } \rho^2 \leq \varepsilon^2 + a \tag{47}$$

and

$$\left\| \left((\varepsilon + i\rho)^2 I + A\right)^{-1} \right\| \leq (2\varepsilon|\rho|)^{-1}, \text{ if } \rho^2 > \varepsilon^2 + a. \tag{48}$$

From (47) we have

$$\left\| \left((\varepsilon + i\rho)^2 I + A\right)^{-1} \right\| \leq \left(\varepsilon^2 + \rho^2 + a\right)^{-1} \left(1 - \frac{a}{\varepsilon^2 + a}\right)^{-1/2} = \frac{\sqrt{\varepsilon^2 + a}}{\varepsilon(\varepsilon^2 + \rho^2 + a)}$$

when $\rho^2 \leq \varepsilon^2 + a$. Therefore, for any ρ we find that

$$\sqrt{\varepsilon^2+\rho^2}\left\|\left((\varepsilon+i\rho)^2 I + A\right)^{-1}\right\| \leq \frac{1}{\varepsilon}.$$

In order to evaluate

$$\left|\xi_m\left((\varepsilon+i\rho)^2\right)\right| = \prod_{j=1}^{m} \frac{h_{j+1,j}}{\left|\left(\left(\varepsilon^2-\rho^2\right)+\mu_j\right)+2i\varepsilon\rho\right)\right|},$$

we notice that

$$\begin{aligned}
\left|\left(\left(\left(\varepsilon^2-\rho^2\right)+\mu_j\right)+2i\varepsilon\rho\right)\right|^2 &= \left(\varepsilon^2+\rho^2\right)^2+\mu_j^2+2\mu_j\left(\varepsilon^2-\rho^2\right) \\
&= \left(\varepsilon^4+\rho^4+2\left(\varepsilon^2-\mu_j\right)\rho^2+\mu_j^2+2\mu_j\varepsilon^2\right) \\
&\geq \left(\varepsilon^4+\rho^4+2\varepsilon^2\rho^2+\mu_j^2+2\mu_j\varepsilon^2\right)\left(1-\frac{\mu_j}{\left(\varepsilon^2+\mu_j\right)}\right) \\
&= \left(\left(\varepsilon^2+\mu_j\right)^2+\rho^2\left(\rho^2+2\varepsilon^2\right)\right)\left(\frac{\varepsilon^2}{\varepsilon^2+\mu_j}\right) \\
&= \varepsilon^2\left(\varepsilon^2+\mu_j\right)\left(1+\frac{\rho^2\left(\rho^2+2\varepsilon^2\right)}{\left(\varepsilon^2+\mu_j\right)^2}\right).
\end{aligned}$$

As a consequence, due to (6) and (45), an upper bound for the error $\|y^{(c)}(t)-y_m^{(c)}(t)\|$ is given by the quantity

$$\begin{aligned}
&\frac{\exp(t\varepsilon)}{\varepsilon\pi}\prod_{j=1}^{m}\frac{h_{j+1,j}}{\varepsilon\left(\varepsilon^2+\mu_j\right)^{1/2}}\int_0^\infty\prod_{j=1}^{m}\left(1+\frac{\rho^2\left(\rho^2+2\varepsilon^2\right)}{\left(\varepsilon^2+\mu_j\right)^2}\right)^{-1/2}d\rho \\
&\leq \frac{\exp(t\varepsilon)}{\varepsilon^{m+1}\pi}\prod_{j=1}^{m}\frac{h_{j+1,j}}{\left(\varepsilon^2+\mu_j\right)^{1/2}}\int_0^\infty\prod_{j=1}^{m}\left(1+\frac{2\rho^2\varepsilon^2}{\left(\varepsilon^2+\mu_j\right)^2}\right)^{-1/2}d\rho \\
&\leq \frac{\exp(t\varepsilon)}{\varepsilon^{m+1}\pi}\prod_{j=1}^{m}\frac{h_{j+1,j}}{\left(\varepsilon^2+\mu_j\right)^{1/2}}\frac{R}{\varepsilon\sqrt{2}}\int_0^\infty\left(1+x^2\right)^{-1}d\rho \\
&\leq \frac{\exp(t\varepsilon)}{\varepsilon^{m+1}\pi}\prod_{j=1}^{m}\frac{h_{j+1,j}}{\left(\varepsilon^2+\mu_j\right)^{1/2}}\frac{R\pi}{2\varepsilon\sqrt{2}},
\end{aligned}$$

which is the desired result. □

Also in this case, in practice further improvement can be reached by choosing the value for ε which minimizes

$$\frac{\exp(t\varepsilon)}{\varepsilon^{m+2}\prod_{j=1}^{m}(\varepsilon^2+\mu_j)^{1/2}}. \tag{49}$$

Moreover, since H_m is symmetric positive definite, $\cos(t\sqrt{H_m})e_1$ can be computed by diagonalization of H_m.

Finally, as what concerns the computation of the *sinc* function

$$y^{(s)}(t) = \left(\sqrt{A}\right)^{-1}\sin\left(t\sqrt{A}\right)v, \quad t>0,$$

we notice that the relationship

$$y^{(s)}(t) = \int_0^t \cos\left(\tau\sqrt{A}\right) v d\tau$$

holds for each $t > 0$ so that, by integrating (46), the error bound

$$\left\| y^{(s)}(t) - y_m^{(s)}(t) \right\| \leq \frac{(\exp(t\varepsilon) - 1)R}{2\sqrt{2}\varepsilon^{m+3}} \prod_{j=1}^{m} \frac{h_{j+1,j}}{(\varepsilon^2 + \mu_j)^{1/2}}$$

is obtained for the corresponding PA approximation.

4. Numerical experiments. In order to provide some numerical results aiming to test the effectiveness of the previous error estimates, we account for some matrices which arise from the discretization of classical partial differential operators. Precisely we consider

$$L = -\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} + c_1 \frac{\partial}{\partial x} + c_2 \frac{\partial}{\partial y}, \quad c_1, c_2 \in \mathbb{R}, \tag{50}$$

where homogeneous Dirichlet boundary conditions are enforced on the unit square $(0,1) \times (0,1)$. The spatial discretization of this partial differential operator is carried out by central differences with uniform steplength $\delta = 1/(n+1)$ along both directions (i.e., $N = n^2$).

We provide several numerical results in order to compare the estimates given in Propositions 6, 8, and 10 with the true error norm and with a priori error estimates already known in the literature (in the case when they are available according to Propositions 3, 4, and 9). We implement bounds (23), (24), and (25) provided in [17] for $\beta = 1$. In the sequel we define the vector v by

$$v = (1, 1, \ldots, 1)^T / n \in \mathbb{R}^N \tag{51}$$

or, as an alternative, by the normalized discretization of the following function

$$v(x, y) = x(1 - x)y(1 - y). \tag{52}$$

Figures 1 and 2 refer to cases concerned with Proposition 6. We account for the φ_0-function applied to the matrix $A = B - \lambda_{\min} I$ where B is symmetric (given by setting $c_1 = c_2 = 0$ and discretizing (50)) and $\lambda_{\min}$ represents its minimum eigenvalue; under this assumption, it is possible to compare estimates (23)–(24), (26), and (37) as it is shown in Figure 1 (on the left) where we have set $n = 50$, $t = 0.005$, and defined v by (52) (normalized). As another test, we consider the skew-symmetric matrix defined as

$$A = \begin{pmatrix} O & B \\ -B & O \end{pmatrix},$$

where B is the symmetric block which arises from the discretization of (50) setting $c_1 = c_2 = 0$ and $n = 30$; in Figure 1 (on the right) we provide the numerical results obtained in correspondence with $\varphi_0(-tA)\overline{v}$ where $t = 0.001$ and $\overline{v} = (v, v)^T / (\|(v, v)\|)$ with v defined by (52). It is evident that, in both cases, our approach is in agreement with the true error norm and it outperforms the a priori bounds given in Propositions 3 and 4.

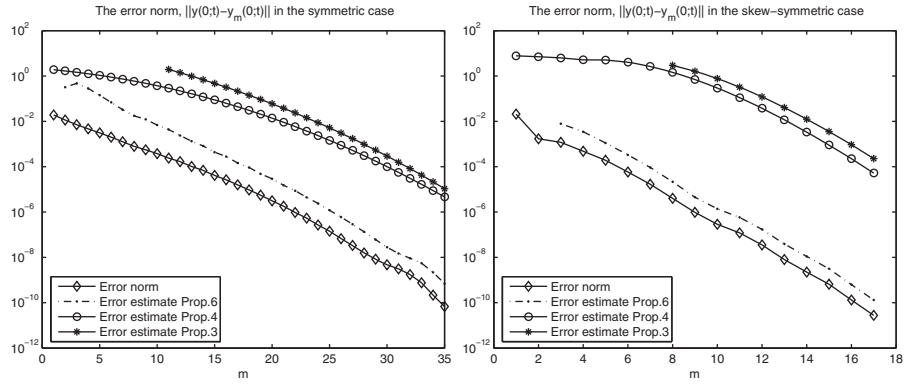


FIG. 1. *The error estimates for the φ_0-function related to Propositions* 3, 4, *and* 6 *compared with the true error norm. On the left: the case when A is symmetric (i.e., $c_1 = c_2 = 0$). On the right: the case when A is skew-symmetric.*

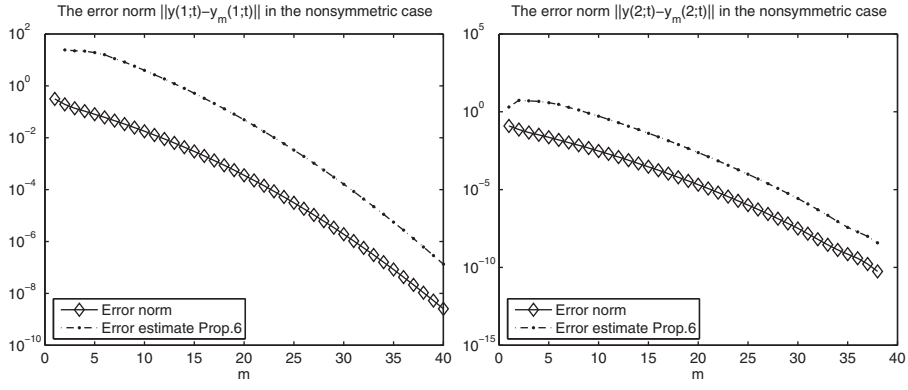


FIG. 2. *The error estimates for the φ_k-function (on the left: $k = 1$; on the right: $k = 2$) related to Proposition* 6 *compared with the true error norm in the case when A is nonsymmetric (i.e., $c_1 = 5$ and $c_2 = 10$).*

Moreover, we are interested in investigating the effectiveness of the estimates we provided concerning the φ_k-functions for $k \geq 1$ in the case of nonsymmetric matrices; for instance, concerning $k = 1, 2$ in Figure 2 we give the results related to the nonsymmetric matrix A obtained by the spatial discretization of (50) where we have set $c_1 = 5$, $c_2 = 10$, $n = 50$, taking $t = 0.005$ and v defined by (51). So far, in all the numerical experiments related to Proposition 6 the parameter ε has been chosen by minimizing quantity (40) except for the skew-symmetric case where ε is set equal to m/t.

In the following, we add further tests: concerning Proposition 8, in Figure 3 we compare estimates (25), (27), and (42) with the true error referring to $\varphi_0(-itA)v$ where $t = 0.001$, v is defined by (52), and A is the symmetric matrix obtained by discretizing (50) with $c_1 = c_2 = 0$, $n = 50$. In Figure 4 we show the results of two experiments concerning $\cos(t\sqrt{A})v$, where we compare the a posteriori estimate of Proposition 10 with the a priori bound of Proposition 9. In both cases we suppose $n = 50$, $t = 0.05$, and v defined by (51). On the left we give the results related

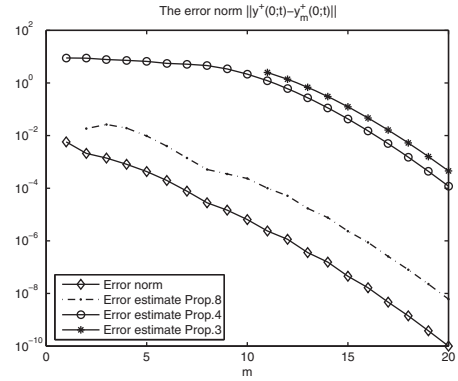


FIG. 3. *The error estimates for the φ_0-function related to Propositions* 3, 4, *and* 8 *compared with the true error norm.*

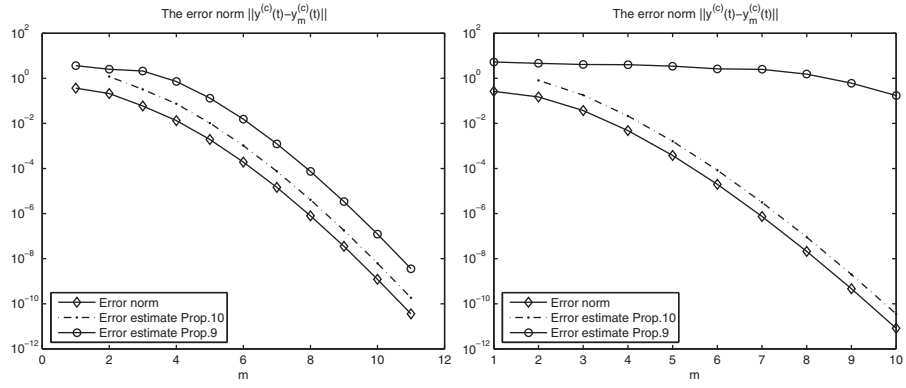


FIG. 4. *The error estimates for the cosine function related to Propositions* 9 *and* 10 *compared with the true error norm.*

to the symmetric matrix A obtained by (50) where $c_1 = c_2 = 0$; while on the right we consider A obtained by using finite differences in order to discretize the following operator

$$L = -\frac{\partial^2}{\partial x^2} - 10\frac{\partial^2}{\partial y^2}$$

on the square $(0,1) \times (0,1)$ with homogeneous Dirichlet conditions in x and homogeneous Neumann conditions in y. In both cases ε is taken by minimizing (49). We notice that in the first case both the estimates match well with the true error; but, in the second test, the a priori bound turns out to be too pessimistic and it is not able to detect the fast convergence of the method, as the a posteriori one does.

We would like to point out that we have found similar results when testing the effectiveness of the proposed estimates on other examples such as the matrices arising from the discretization of the three-dimensional Laplacian operator.

All the numerical experiments have been performed in a MatLab environment, and every value $\varphi(H_m)$ has been computed by means of diagonalization (when it has been possible) or by built-in functions (`expm` or more in general `funm`).

Acknowledgments. The authors would like to thank the referees for advising us to perform comparison with the estimates provided in [7] which helped in the improvement of the numerical part.

REFERENCES

[1] M. Abramovitz and A. Stegun, *Handbook of Mathematical Functions*, Dover Publications Inc., New York, 1965.
[2] H. Berland, B. Skaflestad, and W. Wright, *Expint – A Matlab package for exponential integrators*, ACM Trans. Math. Software, 33 (2007).
[3] D. A. Bini, N. J. Higham, and B. Meini, *Algorithms for the matrix pth root*, Numer. Algorithms, 39 (2005), pp. 349–378.
[4] M. Caliari, M. Vianello, and L. Bergamaschi, *Integrating discrete advection-diffusion propagators at spectral Leja sequence*, J. Comput. Appl. Math., 172 (2004), pp. 79–99.
[5] E. Celledoni and I. Moret, *A Krylov projection method for systems of ODEs*, Appl. Numer. Math., 24 (1997), pp. 365–378.
[6] P. I. Davies and N. Higham, *Computing f(A)v for matrix functions f*, in QCD and Numerical Analysis III, A. Borici, A. Frommer, B. Joo, A. Kennedy, and B. Pendleton, eds., Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, 47 (2005) pp. 15–24.
[7] V. Druskin and L. Knizhnerman, *Two polynomial methods for calculating functions of symmetric matrices*, Comput. Math. Math. Phys., 29 (1989), pp. 112–121.
[8] M. Eiermann and O. G. Ernst, *A restarted Krylov subspace method for the evaluation of matrix functions*, SIAM J. Numer. Anal., 44 (2006), pp. 2481–2504.
[9] K. J. Engel and R. Nagel, *One-Parameter Semigroups for Linear Evolution Equations*, Springer, New York, 2000.
[10] J. v. d. Eshof and M. Hochbruck, *Preconditioning Lanczos approximations to the matrix exponential*, SIAM J. Sci. Comput., (27) 2005, pp. 1438–1457.
[11] H. O. Fattorini, *Second Order Linear Differential Equations in Banach Spaces*, North Holland, Amsterdam, 1995.
[12] A. Frommer and V. Simoncini, *Stopping criteria for rational matrix functions of Hermitian and symmetric matrices*, SIAM J. Sci. Comput., 30 (2008), pp. 1387–1412.
[13] W. Gautschi, *Numerical Analysis, an Introduction*, Birkhäuser, Boston, 1997.
[14] I. Gohberg and S. Goldberg, *Basic Operator Theory*, Birkhäuser, Boston, 1980.
[15] E. Hairer, Ch. Lubich, and G. Wanner, *Geometric Numerical Integration*, Springer, Berlin, 2006.
[16] M. Hochbruck and C. Lubich, *Exponential integrators for quantum-classical molecular dynamics*, BIT, 39 (1999), pp. 620–645.
[17] M. Hochbruck and C. Lubich, *On Krylov subspace approximation to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925.
[18] M. Hochbruck, C. Lubich, and H. Selhofer, *Exponential integrators for large systems of differential equations*, SIAM J. Sci. Comput., 19 (1998), pp. 1552–1574.
[19] M. Hochbruck and A. Ostermann, *Exponential Runge-Kutta methods for parabolic problems*, Appl. Numer. Math., 53 (2005), pp. 323–339.
[20] M. Hochbruck and A. Ostermann, *Explicit exponential Runge-Kutta methods for semilinear parabolic problems*, SIAM J. Numer. Anal., 43 (2005), pp. 1069–1090.
[21] A.-K. Kassam and L. N. Trefethen, *Fourth-order time stepping for stiff PDEs*, SIAM J. Sci. Comput., 26 (2005), pp. 1214–1233.
[22] L. Knizhnermann, *Calculation of functions of unsymmetric matrices using Arnoldi's method*, U.S.S.R. Comput. Maths. Math. Phys., 31 (1991), pp. 1–9, (English Edition by Pergamon Press).
[23] L. Lopez and V. Simoncini, *Analysis of projection methods for rational function approximation to the matrix exponential*, SIAM J. Numer. Anal., 44 (2006), pp. 613–635.
[24] G. Meurant and Z. Strakos, *The Lancsos and conjugate gradients algorithms in finite precision arithmetic*, Acta Numer., 15 (2006), pp. 471–542.
[25] B. V. Minchev and W. M. Wright, *A review of exponential integrators for first order semilinear problems*, preprint Numerics 2/2005, Norwegian University of Science and Technology, Trondheim, Norway, www math.ntnu.no/preprint/numerics/2005/ N2-2005.ps.
[26] I. Moret and P. Novati, *An interpolatory approximation of the matrix exponential based on Faber polynomials*, J. Comput. Appl. Math., 131 (2001), pp. 361–380.

[27] I. Moret and P. Novati, *Interpolating functions of matrices on zeros of quasi-kernel polynomials*, Numer. Linear Algebra Appl., 12 (2005), pp. 337–353.
[28] I. Moret and P. Novati, *RD-rational approximations of the matrix exponential*, BIT, 44 (2004), pp. 595–615.
[29] A. Pazy, *Semigroups of Linear Operators and Applications to Partial Differential Equations*, Springer, New York, 1983.
[30] M. Popolizio and V. Simoncini, *Acceleration techniques for approximating the matrix exponential operator*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 657–683.
[31] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 29 (1992), pp. 209–228.
[32] B. A. Schmitt and R. Weiner, *Matrix-free W-methods using a multiple Arnoldi iteration*, Appl. Numer. Math., 18 (1995), pp. 307–320.
[33] V. Simoncini, *On the convergence of restarted Krylov subspace methods*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 430–452.
[34] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1566–1592

STRUCTURE-PRESERVING ALGORITHMS FOR PALINDROMIC QUADRATIC EIGENVALUE PROBLEMS ARISING FROM VIBRATION OF FAST TRAINS*

TSUNG-MING HUANG†, WEN-WEI LIN‡, AND JIANG QIAN§

Abstract. In this paper, based on Patel's algorithm (1993), we propose a structure-preserving algorithm for solving palindromic quadratic eigenvalue problems (QEPs). We also show the relationship between the structure-preserving algorithm and the URV-based structure-preserving algorithm by Schröder (2007). For large sparse palindromic QEPs, we develop a generalized $\top$-skew-Hamiltonian implicitly restarted shift-and-invert Arnoldi algorithm for solving the resulting $\top$-skew-Hamiltonian pencils. Numerical experiments show that our proposed structure-preserving algorithms perform well on the palindromic QEP arising from a finite element model of high-speed trains and rails.

Key words. palindromic quadratic eigenvalue problem, $\top$-symplectic pencil, $\top$-skew-Hamiltonian pencil

AMS subject classifications. 65F15, 15A18, 15A57

DOI. 10.1137/080713550

1. Introduction.

In this paper, we consider the palindromic quadratic eigenvalue problem (QEP) of the form

$$\mathcal{P}(\lambda)x \equiv (\lambda^2 A_1^\top + \lambda A_0 + A_1)x = 0, \tag{1.1}$$

where $\lambda \in \mathbb{C}$, $x \in \mathbb{C}^n \backslash \{0\}$ and $A_1, A_0 \in \mathbb{C}^{n\times n}$ with $A_0^\top = A_0$. Note that the superscript "$\top$" denotes the complex transpose. The scalar λ and the nonzero vector x in (1.1) are the eigenvalue and the associated eigenvector of $\mathcal{P}(\lambda)$, respectively. The underlying matrix polynomial $\mathcal{P}(\lambda)$ has the property that reversing the order of the coefficients, followed by taking the transpose, leads back to the original matrix polynomial, which explains the word "palindromic." Consequently, taking the transpose of (1.1), we easily see that the eigenvalues of $\mathcal{P}(\lambda)$ satisfy the "symplectic" property; that is, they are paired with respect to the unit circle, containing both an eigenvalue λ and its reciprocal $1/\lambda$ (with 0 and ∞ considered to be reciprocal).

The palindromic QEP (1.1) was first raised in the study of the vibration in the structural analysis for fast trains in Germany [3, 4], associated with the company SFE GmbH in Berlin. Existing fast train systems, like the Japanese Shinkansen, the French TGV, and the German ICE, are being modernized and expanded. Vibration is produced from the interaction between the wheels of a train and the rails underneath. Due to the ever increasing speed (currently up to 300 km/hr) of modern trains, the study of its vibration becomes an important task. Research does not only contribute

*Received by the editors January 17, 2008; accepted for publication (in revised form) by F. Tisseur September 3, 2008; published electronically January 16, 2009.
http://www.siam.org/journals/simax/30-4/71355.html

†Department of Mathematics, National Taiwan Normal University, Taipei, 116, Taiwan (min@math.ntnu.edu.tw). This author's work was partially supported by the National Science Council and the National Center for Theoretical Sciences in Taiwan.

‡Department of Applied Mathematics, National Chiao-Tung University, Hsinchu, 300, Taiwan (wwlin@math.nctu.edu.tw).

§School of Sciences, Beijing University of Posts and Telecommunications, Beijing, 100876, China; Department of Mathematics, National Tsinghua University, Hsinchu, 300, Taiwan (jqian104@gmail.com). This author's research was partly supported by Project NSFC 10571007.

towards the increased comfort of passengers, in terms of lower noise and vibration levels. More importantly, the safety in the operation of the trains will be improved, and the operational and construction costs will be optimized [4, 5, 12, 13]. In addition, innovative designs of railway bridges, embedded rail structures, and train suspension systems require accurate resolution of the vibration.

A standard approach for solving the palindromic QEP (1.1) is to transform it into a $2n \times 2n$ linear eigenvalue problem

$$\begin{bmatrix} 0 & I \\ A_1 & A_0 \end{bmatrix} \begin{bmatrix} x \\ \lambda x \end{bmatrix} = \lambda \begin{bmatrix} I & 0 \\ 0 & -A_1^\top \end{bmatrix} \begin{bmatrix} x \\ \lambda x \end{bmatrix} \tag{1.2}$$

and compute its generalized Schur form (see [23]). However, the symplectic property of the eigenvalues of (1.1) is not preserved by computation, generally, producing large numerical errors ([5]). Recently, some pioneering work [4, 12, 13] proposed a good linearization which linearizes the palindromic QEP (1.1) into the form $\lambda Z^\top + Z$, which preserves symplecticity to some extent, and suggested some structure-preserving solution methods. This leads to a vast improvement over previous approaches. Later, a QR-like algorithm [19] and a Jacobi-type method [4] combined with the Laub trick, a preprocessing step of the generalized Schur form [11], have been developed for solving the palindromic linear pencil $\lambda Z^\top + Z$. However, the latter method works well, only if there are no eigenvalues near ± 1. The Jacobi method typically needs about $O(n^3 \log(n))$ flops and the QR-like algorithm is of $O(n^4)$ flops. Recently, a URV-decomposition-based structured method of cubic complexity was developed in [20] to solve the palindromic linear pencil $\lambda Z^\top + Z$, producing eigenvalues which are paired to working precision. In section 3, we will show that the URV-based method [20] is mathematically equivalent to applying the structure-preserving algorithm in section 2 to the enlarged $2n \times 2n$ palindromic quadratic pencil $\zeta^2 Z^\top + \zeta 0 + Z$ (with $\zeta^2 = \lambda$). On the other hand, a structure-preserving doubling algorithm was developed in [1] via the computation of a solvent of a nonlinear matrix equation associated with (1.1). The numerical results show much promise but the convergence theory holds only when the algorithm does not break down.

As mentioned before, the linearization (1.2) generally cannot preserve the symplectic structure. Fortunately, the special linearization for (1.1) (see [1] or [10])

$$(\mathcal{M} - \lambda \mathcal{L})z \equiv \left(\begin{bmatrix} A_1 & 0 \\ -A_0 & -I \end{bmatrix} - \lambda \begin{bmatrix} 0 & I \\ A_1^\top & 0 \end{bmatrix} \right) \begin{bmatrix} x \\ y \end{bmatrix} = 0 \tag{1.3}$$

obtained by setting $y = \frac{1}{\lambda} A_1 x$ and multiplying the second equation of (1.3) by λ satisfies

$$\mathcal{M} \mathcal{J} \mathcal{M}^\top = \mathcal{L} \mathcal{J} \mathcal{L}^\top, \tag{1.4}$$

where $\mathcal{J} \equiv \mathcal{J}_{2n}$ is the $2n \times 2n$ matrix $\left[\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix} \right]$. In other words, the pencil $\mathcal{M} - \lambda \mathcal{L}$ or the matrix pair $(\mathcal{M}, \mathcal{L})$ in (1.3) preserves the symplectic structure of (1.4) and is said to be $\top$-symplectic.

For a real matrix pair $(\mathcal{M}, \mathcal{L})$ satisfying (1.4), a structure-preserving $(\mathcal{S} + \mathcal{S}^{-1})$-transform for the computation of all its eigenvalues is proposed by [9] and a numerically stable algorithm for reducing the transformed pair to a block triangular condensed form by using only orthogonal transformations was developed by Patel [16]. It is perfectly suitable for the $\top$-symplectic pair, but not applicable to the complex conjugate

symplectic pair (i.e., $\mathcal{M}\mathcal{J}\mathcal{M}^H = \mathcal{L}\mathcal{J}\mathcal{L}^H$). In this paper, we adapt Patel's approach to solve the $\top$-symplectic pencil in (1.3) resulting from the palindromic QEP (1.1). Only unitary transformations are used and the symplectic structure is fully preserved, which make the method attractive. It is worth mentioning that the $(\mathcal{S}+\mathcal{S}^{-1})$-transform is, in general, a nonlinear transform as in solving the discrete-time optimal control problem [9, 16]. However, the special form in (1.3) leads to a linear $(\mathcal{S}+\mathcal{S}^{-1})$-transform without involving any matrix multiplication.

In some applications, the matrices A_1 and A_0 in (1.1) (and hence $\mathcal{M}$ and $\mathcal{L}$ in (1.3)) can be large and sparse and only the eigenvalues in a specified region are required. To accomplish this, the shift-and-invert (implicitly restarted) Arnoldi algorithm [7, 17, 21] is one of the most widely used standard techniques for computing selected eigenvalues of the large sparse matrix pencil $\mathcal{M}-\lambda\mathcal{L}$. In this approach, the corresponding shifted and inverted matrix is reduced to a Hessenberg form which no longer has the desirable symplectic structure.

Mehrmann and Watkins [15] developed a structure-preserving skew-Hamiltonian, isotropic, implicitly restarted shift-and-invert Arnoldi algorithm (SHIRA) for the computation of eigenpairs of a large sparse real skew-Hamiltonian/Hamiltonian pencil by transforming the pencils to a skew-Hamiltonian operator. In fact, SHIRA can be straightforwardly extended to solve a skew-Hamiltonian/Hamiltonian pencil in the complex transpose case (not in the complex conjugate case), referred to as $\top$SHIRA. We first transform the $\top$-symplectic pencil to a $\top$-skew-Hamiltonian eigenvalue problem by using the $(\mathcal{S}+\mathcal{S}^{-1})$-transform, then $\top$SHIRA is applied to the resulting $\top$-skew-Hamiltonian matrix. On the other hand, to avoid explicitly forming the $\top$-skew-Hamiltonian matrix in the above transformation, we also develop a generalized $\top$-skew-Hamiltonian implicitly restarted shift-and-invert Arnoldi algorithm (G$\top$SHIRA) for solving the $\top$-skew-Hamiltonian pencil resulting from the $(\mathcal{S}+\mathcal{S}^{-1})$-transform of the symplectic pencil $\mathcal{M}-\lambda\mathcal{L}$.

We introduce some definitions that will be used frequently in this paper.

DEFINITION 1.1.

(i) *A matrix $\mathcal{A}\in\mathbb{C}^{n\times n}$ is called $\top$-symmetric or $\top$-skew-symmetric if it satisfies $\mathcal{A}^\top=\mathcal{A}$ or $\mathcal{A}^\top=-\mathcal{A}$, respectively.*

(ii) *A matrix $\mathcal{U}\in\mathbb{C}^{2n\times 2n}$ is called $\top$-symplectic if $\mathcal{U}^\top\mathcal{J}\mathcal{U}=\mathcal{J}$; a pencil $\mathcal{M}-\lambda\mathcal{L}\in\mathbb{C}^{2n\times 2n}$ or the matrix pair $(\mathcal{M},\mathcal{L})$ is called $\top$-symplectic if $\mathcal{M}\mathcal{J}\mathcal{M}^\top=\mathcal{L}\mathcal{J}\mathcal{L}^\top$.*

(iii) *A matrix $\mathcal{H}\in\mathbb{C}^{2n\times 2n}$ is called $\top$-Hamiltonian or $\top$-skew-Hamiltonian if it satisfies $(\mathcal{H}\mathcal{J})^\top=\mathcal{H}\mathcal{J}$ or $(\mathcal{H}\mathcal{J})^\top=-\mathcal{H}\mathcal{J}$, respectively.*

(iv) *A pencil $\mathcal{K}-\lambda\mathcal{N}\in\mathbb{C}^{2n\times 2n}$ or the matrix pair $(\mathcal{K},\mathcal{N})$ is called $\top$-skew-Hamiltonian if $\mathcal{K}$ and $\mathcal{N}$ are $\top$-skew-Hamiltonian.*

(v) *Let $X,Y\in\mathbb{C}^{2n\times m}(1\le m\le n)$; X is called $\top$-isotropic if $X^\top\mathcal{J}X=0_m$; and X and Y are called $\top$-bi-isotropic if $X^\top\mathcal{J}Y=0_m$.*

Throughout this paper, $A^\top$ and A^H denote the transpose and conjugate transpose of a matrix A, respectively. We denote the $m\times n$ zero matrix by $0_{m,n}$, and the zero and identity matrices of order n by 0_n and I_n, respectively. The ith column of I_n is denoted by e_i. We adopt the following MATLAB notations: $v(i:j)$ denotes the subvector of the vector v that consists of the ith to the jth entries of v. $A(i:j,k:\ell)$ denotes the submatrix of the matrix A that consists of the intersection of the rows i to j and the columns k to ℓ. $A(i:j,:)$ and $A(:,k:\ell)$ select the rows i to j and the columns k to ℓ, respectively, of A.

The paper is organized as follows. In section 2, we briefly present the structure-preserving algorithm based on Patel's method [16] for solving palindromic QEPs. In

section 3, we show the relationship between the structure-preserving algorithm and the URV-based structured method proposed by Schröder [20]. In section 4, based on the SHIRA developed in [15], we introduce the $\top$-skew-Hamiltonian implicitly-restarted shift-and-invert Arnoldi algorithm ($\top$SHIRA) for solving the resulting $\top$-skew-Hamiltonian matrix. In section 5, a generalized $\top$-skew-Hamiltonian implicitly-restarted shift-and-invert Arnoldi algorithm (G$\top$SHIRA) for solving the resulting $\top$-skew-Hamiltonian pencils is developed. We present some numerical results of the proposed algorithms, using examples from a finite element model of fast trains [1], in section 6. Conclusions are given in section 7.

2. Structure-preserving algorithm I. We adapt Patel's algorithm [16] applying to the $(\mathcal{S}+\mathcal{S}^{-1})$-transform of a $\top$-symplectic matrix pair for the computation of all its eigenpairs. Let $(\mathcal{M},\mathcal{L})$ be a $\top$-symplectic pair. The $(\mathcal{S}+\mathcal{S}^{-1})$-transform $(\mathcal{M}_s,\mathcal{L}_s)$ of $(\mathcal{M},\mathcal{L})$ is defined by (see [9])

$$\mathcal{M}_s \equiv \mathcal{M}\mathcal{J}\mathcal{L}^\top + \mathcal{L}\mathcal{J}\mathcal{M}^\top, \qquad \mathcal{L}_s \equiv \mathcal{L}\mathcal{J}\mathcal{L}^\top. \tag{2.1}$$

We first give the relationship between eigenpairs of a $\top$-symplectic pencil and its $(\mathcal{S}+\mathcal{S}^{-1})$-transform.

THEOREM 2.1. *Let $(\mathcal{M},\mathcal{L})$ be a $\top$-symplectic pair and $(\mathcal{M}_s,\mathcal{L}_s)$ be its $(\mathcal{S}+\mathcal{S}^{-1})$-transform. Then*

(i) *μ is a double eigenvalue of $(\mathcal{M}_s,\mathcal{L}_s)$ if and only if $\nu,\frac{1}{\nu}$ are eigenvalues of $(\mathcal{M},\mathcal{L})$, where $\nu,\frac{1}{\nu}$ are two roots of the quadratic equation $\lambda+\frac{1}{\lambda}=\mu$.*

(ii) *Let x and y be linearly independent eigenvectors of $(\mathcal{L}^\top,\mathcal{M}^\top)$ corresponding to ν and $\frac{1}{\nu}$, respectively, i.e., $(\mathcal{L}^\top-\nu\mathcal{M}^\top)x=0$ and $(\mathcal{L}^\top-\frac{1}{\nu}\mathcal{M}^\top)y=0$. Then x and y are two linearly independent eigenvectors of $(\mathcal{M}_s,\mathcal{L}_s)$ corresponding to $\mu=\nu+\frac{1}{\nu}$.*

(iii) *Furthermore, from* (ii)*, if $z_s=\alpha x+\beta y$ (with $\alpha\beta\neq 0$) is an eigenvector of $(\mathcal{M}_s,\mathcal{L}_s)$ corresponding to $\mu=\nu+\frac{1}{\nu}$ ($\mu\neq\pm 2$), i.e., $(\mathcal{M}_s-\mu\mathcal{L}_s)z_s=0$, then $\mathcal{J}(\mathcal{L}^\top-\frac{1}{\nu}\mathcal{M}^\top)z_s$ and $\mathcal{J}(\mathcal{L}^\top-\nu\mathcal{M}^\top)z_s$ are the eigenvectors of $(\mathcal{M},\mathcal{L})$ corresponding to ν and $\frac{1}{\nu}$, respectively.*

Proof.

(i) As in [9], since $\mathcal{M}\mathcal{J}\mathcal{M}^\top=\mathcal{L}\mathcal{J}\mathcal{L}^\top$, by (2.1) it holds that

$$\begin{aligned}\mathcal{M}_s-\mu\mathcal{L}_s &= \mathcal{M}\mathcal{J}\mathcal{L}^\top+\mathcal{L}\mathcal{J}\mathcal{M}^\top-\left(\nu+\frac{1}{\nu}\right)\mathcal{L}\mathcal{J}\mathcal{L}^\top\\ &= (\mathcal{M}-\nu\mathcal{L})\mathcal{J}\left(\mathcal{L}^\top-\frac{1}{\nu}\mathcal{M}^\top\right)\\ &= \left(\mathcal{M}-\frac{1}{\nu}\mathcal{L}\right)\mathcal{J}(\mathcal{L}^\top-\nu\mathcal{M}^\top).\end{aligned} \tag{2.2}$$

Hence (i) follows.

(ii) From the last two equations of (2.2), it follows that

$$(\mathcal{M}_s-\mu\mathcal{L}_s)\,x=\left(\mathcal{M}-\frac{1}{\nu}\mathcal{L}\right)\mathcal{J}(\mathcal{L}^\top-\nu\mathcal{M}^\top)\,x=0,$$

and

$$(\mathcal{M}_s-\mu\mathcal{L}_s)\,y=(\mathcal{M}-\nu\mathcal{L})\mathcal{J}\left(\mathcal{L}^\top-\frac{1}{\nu}\mathcal{M}^\top\right)y=0.$$

(iii) By applying the last two equations of (2.2) again, it remains to show only that $\mathcal{J}(\mathcal{L}^\top - \frac{1}{\nu}\mathcal{M}^\top)z_s \neq 0$ and $\mathcal{J}(\mathcal{L}^\top - \nu\mathcal{M}^\top)z_s \neq 0$. From (ii) we have

$$\mathcal{J}\left(\mathcal{L}^\top - \frac{1}{\nu}\mathcal{M}^\top\right)z_s = \mathcal{J}\left(\mathcal{L}^\top - \frac{1}{\nu}\mathcal{M}^\top\right)(\alpha x + \beta y) = \alpha\mathcal{J}\left(\mathcal{L}^\top - \frac{1}{\nu}\mathcal{M}^\top\right)x \neq 0.$$

Similarly,

$$\mathcal{J}(\mathcal{L}^\top - \nu\mathcal{M}^\top)z_s = \mathcal{J}(\mathcal{L}^\top - \nu\mathcal{M}^\top)(\alpha x + \beta y) = \beta\mathcal{J}(\mathcal{L}^\top - \nu\mathcal{M}^\top)y \neq 0. \qquad \square$$

THEOREM 2.2. *Let $(\mathcal{M}, \mathcal{L})$ be the $\top$-symplectic pair as in (1.3) and $(\mathcal{M}_s, \mathcal{L}_s)$ be its $(\mathcal{S} + \mathcal{S}^{-1})$-transform. If $z_s = [z_1^\top, z_2^\top]^\top$ with $z_1, z_2 \in \mathbb{C}^n$ is an eigenvector of $(\mathcal{M}_s, \mathcal{L}_s)$ corresponding to $\mu = \nu + \frac{1}{\nu}$ $(\mu \neq \pm 2)$, then $z_1 + \frac{1}{\nu}z_2$ and $z_1 + \nu z_2$ are eigenvectors of $\mathcal{P}(\lambda)$ in (1.1) corresponding to ν and $\frac{1}{\nu}$, respectively.*

Proof. From (iii) of Theorem 2.1 we compute

(2.3)
$$\left(\mathcal{J}\left(\mathcal{L}^\top - \frac{1}{\nu}\mathcal{M}^\top\right)z_s\right)(1:n) = z_1 + \frac{1}{\nu}z_2, \quad (\mathcal{J}(\mathcal{L}^\top - \nu\mathcal{M}^\top)z_s)(1:n) = z_1 + \nu z_2.$$

Then, from (1.3) and (2.3), it follows that $\mathcal{P}(\nu)(z_1 + \frac{1}{\nu}z_2) = 0$ and $\mathcal{P}(\frac{1}{\nu})(z_1 + \nu z_2) = 0$. $\square$

Note that from (1.3), we have

$$\begin{aligned}(\mathcal{M}_s, \mathcal{L}_s) &= (\mathcal{M}\mathcal{J}\mathcal{L}^\top + \mathcal{L}\mathcal{J}\mathcal{M}^\top, \mathcal{L}\mathcal{J}\mathcal{L}^\top) \\ &= \left(\begin{bmatrix} A_1 - A_1^\top & A_0 \\ -A_0 & A_1 - A_1^\top \end{bmatrix}, \begin{bmatrix} 0 & -A_1 \\ A_1^\top & 0 \end{bmatrix}\right) \\ &= \left(\begin{bmatrix} A_0 & A_1^\top - A_1 \\ A_1 - A_1^\top & A_0 \end{bmatrix}, \begin{bmatrix} -A_1 & 0 \\ 0 & -A_1^\top \end{bmatrix}\right)\mathcal{J} \\ &\equiv (\mathcal{K}, \mathcal{N})\mathcal{J}. \qquad (2.4)\end{aligned}$$

From (2.4), if z is an eigenvector of $(\mathcal{K}, \mathcal{N})$ corresponding to μ, then $z_s = \mathcal{J}^\top z$ is the eigenvector of $(\mathcal{M}_s, \mathcal{L}_s)$ corresponding to the same μ.

Remark 2.1.

(i) The $(\mathcal{S} + \mathcal{S}^{-1})$-transform $(\mathcal{M}_s, \mathcal{L}_s)$ in (2.1) of a $\top$-symplectic pair, in general, is a nonlinear (quadratic) transformation. For instance, the $(\mathcal{S} + \mathcal{S}^{-1})$-transform of the symplectic pair of the form $(\mathcal{M}, \mathcal{L}) \equiv ([\begin{smallmatrix} A & 0 \\ -H & I \end{smallmatrix}], [\begin{smallmatrix} I & G \\ 0 & A^\top \end{smallmatrix}])$ with $H = H^\top$ and $G = G^\top$ arisen from discrete-time optimal control problems produces a quadratic $(\mathcal{S} + \mathcal{S}^{-1})$-transform which involves matrix multiplications and is not backward stable. However, the special form of the $\top$-symplectic pair $(\mathcal{M}, \mathcal{L})$ in (1.3) leads to a linear $(\mathcal{S} + \mathcal{S}^{-1})$-transform as in (2.4) and does not involve any matrix multiplication.

(ii) The eigenvectors of $\mathcal{P}(\lambda)$ corresponding to ν and $1/\nu$ can be obtained from the eigenvectors of $(\mathcal{K}, \mathcal{N})$ directly (see Theorem 2.2), not requiring us to solve any linear system or perform any matrix-vector multiplications.

It is easily seen that $\mathcal{K}$ and $\mathcal{N}$ in (2.4) are both $\top$-skew-Hamiltonian. Patel [16] introduced two types of transformations that preserve the skew-Hamiltonian structure. The first type involves similarity transformations on $\mathcal{K}$ and $\mathcal{N}$, respectively, using Given rotations $G_0(i, c, \bar{s}) := G(i, n + i, c, \bar{s})$. The second type involves equivalence transformations on $\mathcal{K}$ and $\mathcal{N}$, respectively, by the left transformation $Q_0^\top := (U^\top \oplus V^\top)$ and the right transformation $Z_0 := (V \oplus U)$, where the unitary

$U, V \in \mathbb{C}^{n\times n}$ represent the application of Givens rotations. One can easily verify that the new transforming $\mathcal{K}$ and $\mathcal{N}$ are still $\top$-skew-Hamiltonian.

Based on Patel's approach [16] with these two types of transformations, we may reduce $(\mathcal{K}, \mathcal{N})$ to a block triangular structure; that is,

$$\mathcal{K} := Q^\top \mathcal{K} Z = \begin{bmatrix} K_{11} & K_{12} \\ 0 & K_{11}^\top \end{bmatrix}, \qquad \mathcal{N} := Q^\top \mathcal{N} Z = \begin{bmatrix} N_{11} & N_{12} \\ 0 & N_{11}^\top \end{bmatrix}, \tag{2.5}$$

where $K_{11} \in \mathbb{C}^{n\times n}$ is upper Hessenberg, $N_{11} \in \mathbb{C}^{n\times n}$ is upper triangular, and Q, Z are unitary satisfying

$$Q = \mathcal{J}^\top Z \mathcal{J}. \tag{2.6}$$

From (2.5), we see that the pair (K_{11}, N_{11}) contains half of the eigenvalues of $(\mathcal{K}, \mathcal{N})$. We then apply the QZ algorithm to (K_{11}, N_{11}) for computing all eigenpairs $\{(\mu_i, y_i)\}_{i=1}^n$. Consequently, $\{(\mu_i, Z[\begin{smallmatrix} y_i \\ 0 \end{smallmatrix}])\}_{i=1}^n$ are n eigenpairs of $(\mathcal{K}, \mathcal{N})$. From (2.4), $\{(\mu_i, z_i(\equiv \mathcal{J}^\top Z[\begin{smallmatrix} y_i \\ 0 \end{smallmatrix}]))\}_{i=1}^n$ are eigenpairs of $(\mathcal{M}_s, \mathcal{L}_s)$. Finally, we compute all eigenvalues and the associated eigenvectors of $\mathcal{P}(\lambda)$ by Theorem 2.2.

ALGORITHM 2.1 (structure-preserving algorithm I (SA_I)).

Input: *A palindromic quadratic pencil* $\mathcal{P}(\lambda) \equiv \lambda^2 A_1^\top + \lambda A_0 + A_1$ *with* $A_0, A_1 \in \mathbb{C}^{n\times n}$ *and* $A_0^\top = A_0$.

Output: *All eigenvalues and eigenvectors of* $\mathcal{P}(\lambda)$.

Step 1. *Form the pair* $(\mathcal{K}, \mathcal{N})$ *as in* (2.4)*;*

Step 2. *Reduce* $(\mathcal{K}, \mathcal{N})$ *to block upper triangular forms in* (2.5) *using unitary transformations. (See a pseudocode in Appendix* A.1.*);*

Step 3. *Compute eigenpairs* $\{(\mu_i, y_i)\}_{i=1}^n$ *of* (K_{11}, N_{11}) *defined in* (2.5) *by using the QZ algorithm;*

Step 4. *Compute* $z_i = \mathcal{J}^\top Z[\begin{smallmatrix} y_i \\ 0 \end{smallmatrix}] \equiv [\begin{smallmatrix} z_{i1} \\ z_{i2} \end{smallmatrix}], i = 1, 2, \ldots, n;$

Step 5. *Compute eigenvalues* ν_i *and* $\frac{1}{\nu_i}$ *of* $\mathcal{P}(\lambda)$ *by solving* $\nu^2 - \mu_i \nu + 1 = 0$*; Compute eigenvectors* $x_{i1} \equiv z_{i1} + \frac{1}{\nu_i} z_{i2}$, $x_{i2} \equiv z_{i1} + \nu_i z_{i2}$ *corresponding to* $\nu_i, \frac{1}{\nu_i}$, *respectively, for* $i = 1, 2, \ldots, n$.

Remark 2.2. The SA_I requires approximately $27n^3$ flops for the eigenvalues, and an additional $23n^3$ flops for the eigenvectors. While the QZ algorithm is applied to $(\mathcal{M}, \mathcal{L})$ directly, it requires approximately $120n^3$ flops for the eigenvalues and an additional $\frac{260}{3}n^3$ flops for the eigenvectors. Here and hereafter a flop is a floating point multiplication and addition for complex numbers, which involves 6 real flops.

3. Structure-preserving algorithm II vs. URV-based method. Recently in [4, 12, 13], a "good" linearization of the palindromic quadratic pencil (1.1) was proposed:

$$\lambda Z^\top + Z \equiv \lambda \begin{bmatrix} A_1^\top & A_0 - A_1 \\ A_1^\top & A_1^\top \end{bmatrix} + \begin{bmatrix} A_1 & A_1 \\ A_0 - A_1^\top & A_1 \end{bmatrix}. \tag{3.1}$$

This preserves the "symplecticity" of the eigenvalues. In order to solve the palindromic linear eigenvalue problem of (3.1), we rewrite it into a new palindromic quadratic pencil

$$\mathcal{Q}(\zeta) \equiv \zeta^2 Z^\top + \zeta 0_{2n} + Z \tag{3.2}$$

with $\zeta^2 = \lambda$. We then apply the SA_I algorithm proposed in section 2 to solve the palindromic QEP of (3.2). As in (2.4), we form

$$\widetilde{\mathcal{K}} = \begin{bmatrix} 0 & Z^\top - Z \\ Z - Z^\top & 0 \end{bmatrix}, \quad \widetilde{\mathcal{N}} = \begin{bmatrix} -Z & 0 \\ 0 & -Z^\top \end{bmatrix}. \tag{3.3}$$

By (2.5) there are unitary $\mathcal{U}_a, \mathcal{V}_a \in \mathbb{C}^{4n\times 4n}$ with $\mathcal{U}_a = \mathcal{J}_{4n}^\top \mathcal{V}_a \mathcal{J}_{4n}$ such that

$$\mathcal{U}_a^\top \widetilde{\mathcal{K}} \mathcal{V}_a = \begin{bmatrix} K_{11}^a & K_{12}^a \\ 0 & (K_{11}^a)^\top \end{bmatrix}, \quad \mathcal{U}_a^\top \widetilde{\mathcal{N}} \mathcal{V}_a = \begin{bmatrix} N_{11}^a & N_{12}^a \\ 0 & (N_{11}^a)^\top \end{bmatrix}, \tag{3.4}$$

where $K_{11}^a \in \mathbb{C}^{2n\times 2n}$ is upper Hessenberg with $\{0, 2, \ldots, 2n-2\}$-diagonals being zeros, $N_{11}^a \in \mathbb{C}^{2n\times 2n}$ is upper triangular with $\{1, 3, \ldots, 2n-1\}$-diagonals being zeros, and K_{12}^a and $N_{12}^a \in \mathbb{C}^{2n\times 2n}$ are skew symmetric with $\{1, -1, \ldots, 2n-1, -(2n-1)\}$-diagonals and with $\{2, -2, \ldots, 2n-2, -(2n-2)\}$-diagonals, respectively, being zeros. Here the ℓ-diagonal of a matrix $A \equiv [a_{ij}]_{i,j=1}^n$ consists of the entries $\{a_{ij}\}$ with $j - i = \ell$. Note that the extra zeros in K_{11}^a, N_{11}^a, K_{12}^a, and N_{12}^a are obtained by performing some suitable permutations on the special forms of (3.3) without any calculation. (See Appendix A.2 for details.) Denote

$$\mathcal{P}_{2n} = [e_1, e_{n+1}, e_2, e_{n+2}, \ldots, e_n, e_{2n}]. \tag{3.5}$$

Let

$$\mathcal{U}^\top = \begin{bmatrix} \mathcal{P}_{2n}^\top & 0 \\ 0 & \mathcal{P}_{2n}^\top \end{bmatrix} \mathcal{U}_a^\top, \quad \mathcal{V} = \mathcal{V}_a \begin{bmatrix} \mathcal{P}_{2n} & 0 \\ 0 & \mathcal{P}_{2n} \end{bmatrix}. \tag{3.6}$$

Then we have

$$\mathcal{U}^\top \widetilde{\mathcal{K}} \mathcal{V} = \left[\begin{array}{cc|cc} 0 & R_1 & T_1 & 0 \\ R_2 & 0 & 0 & -T_2 \\ \hline 0 & 0 & 0 & R_2^\top \\ 0 & 0 & R_1^\top & 0 \end{array}\right], \tag{3.7a}$$

$$\mathcal{U}^\top \widetilde{\mathcal{N}} \mathcal{V} = \left[\begin{array}{cc|cc} R_3 & 0 & 0 & -T_3 \\ 0 & R_4 & T_3^\top & 0 \\ \hline 0 & 0 & R_3^\top & 0 \\ 0 & 0 & 0 & R_4^\top \end{array}\right], \tag{3.7b}$$

where $R_1 \in \mathbb{C}^{n\times n}$ is upper Hessenberg, $R_2, R_3, R_4 \in \mathbb{C}^{n\times n}$ are upper triangular, $T_1, T_2 \in \mathbb{C}^{n\times n}$ are skew symmetric, and $T_3 \in \mathbb{C}^{n\times n}$. From (3.7), we see that in order to compute the eigenvalues and the eigenvectors of $(\widetilde{\mathcal{K}}, \widetilde{\mathcal{N}})$ it suffices to compute those of the matrix pair

$$(R_1 R_4^{-1} R_2, R_3). \tag{3.8}$$

We apply the periodic QZ algorithm [2, 18] to the matrix pair in (3.8) without forming the product explicitly, which gives the n eigenpairs $\{(\gamma_i, y_i)\}_{i=1}^n$, where $y_i \in \mathbb{C}^n$. Let $\mu_i = \sqrt{\gamma_i}$ (one branch of the square root of γ_i), $\eta_i := \mu_i R_1^{-1} R_3 y_i$, and $\widetilde{y}_i = [y_i^\top, \eta_i^\top]^\top$. It follows that $\{(\mu_i, \widetilde{z}_i (\equiv \mathcal{V}\begin{bmatrix} \widetilde{y}_i \\ 0 \end{bmatrix}))\}_{i=1}^n$ are n eigenpairs of $(\widetilde{\mathcal{K}}, \widetilde{\mathcal{N}})$. Write $\widetilde{z}_i = [\widetilde{z}_{i1}^\top, \widetilde{z}_{i2}^\top]^\top$ and solve ν_i and $\frac{1}{\nu_i}$ for $\nu^2 + (2 - \mu_i^2)\nu + 1 = 0$. By Theorem 2.2 and (3.1), we compute the eigenvectors

$$x_{i1} = \widetilde{x}_{i1}(1:n) + \widetilde{x}_{i1}(n+1:2n), \quad x_{i2} = \widetilde{x}_{i2}(1:n) + \widetilde{x}_{i2}(n+1:2n) \tag{3.9a}$$

of $\mathcal{P}(\lambda)$ corresponding to ν_i and $\frac{1}{\nu_i}$, respectively, where

$$\widetilde{x}_{i1} := \widetilde{z}_{i2} - \frac{1}{\sqrt{\nu_i}}\widetilde{z}_{i1}, \quad \widetilde{x}_{i2} := \widetilde{z}_{i2} - \sqrt{\nu_i}\widetilde{z}_{i1}. \tag{3.9b}$$

ALGORITHM 3.1 (structure-preserving algorithm II (SA_II)).

Input: *A palindromic quadratic pencil* $\mathcal{P}(\lambda) \equiv \lambda^2 A_1^\top + \lambda A_0 + A_1$ *with* $A_0, A_1 \in \mathbb{C}^{n\times n}$ *and* $A_0^\top = A_0$.

Output: *All eigenvalues and eigenvectors of* $\mathcal{P}(\lambda)$.

Step 1. *Form the pair* $(\widetilde{\mathcal{K}}, \widetilde{\mathcal{N}})$ *as in* (3.3);

Step 2. *Reduce* $(\widetilde{\mathcal{K}}, \widetilde{\mathcal{N}})$ *to block upper triangular forms as in* (3.7) *using unitary transformations of* (3.4)–(3.6);

Step 3. *Compute eigenpairs* $\{(\gamma_i, y_i)\}_{i=1}^n$ *of* $(R_1 R_4^{-1} R_2, R_3)$ *in* (3.8) *by the periodic QZ algorithm* [18];

Step 4. *Compute* $\widetilde{z}_i = \mathcal{V}\left[\begin{smallmatrix}\widetilde{y}_i\\ 0\end{smallmatrix}\right] \equiv \left[\begin{smallmatrix}\widetilde{z}_{i1}\\ \widetilde{z}_{i2}\end{smallmatrix}\right]$, *where* $\widetilde{y}_i = \left[\begin{smallmatrix} I_n \\ \sqrt{\gamma_i} R_1^{-1} R_3\end{smallmatrix}\right] y_i$ *for* $i = 1, 2, \ldots, n$;

Step 5. *Compute* ν_i *and* $\frac{1}{\nu_i}$ *by solving* $\nu^2 + (2 - \gamma_i)\nu + 1 = 0$; *Compute eigenvectors* x_{i1} *and* x_{i2} *of* $\mathcal{P}(\lambda)$ *as in* (3.9a) *corresponding to* $\nu_i, \frac{1}{\nu_i}$, *respectively, for* $i = 1, 2, \ldots, n$.

Remark 3.1.

(i) In Step 3, since R_1, R_4, R_2, and R_3 are already in Hessenberg-triangular form, the first step in the periodic QZ algorithm is not needed.

(ii) The SA_II requires $62n^3$ flops for the eigenvalues, and an additional $23n^3$ flops for the eigenvectors.

Recently a URV-decomposition-based structured method was proposed in [20] for solving the palindromic linear pencil (3.1). From [20] there are unitary $U, V \in \mathbb{C}^{2n\times 2n}$ such that

$$U^\top Z V = \begin{bmatrix} 0 & \widehat{R}_4^\top \Pi_n \\ \Pi_n \widehat{R}_3 & \Pi_n \widehat{T}_3 \Pi_n \end{bmatrix}, \quad V^\top (Z - Z^\top) V = \begin{bmatrix} 0 & -\widehat{R}_2^\top \Pi_n \\ \Pi_n \widehat{R}_2 & \Pi_n \widehat{T}_2 \Pi_n \end{bmatrix} \tag{3.10a}$$

and

$$U^\top (Z^\top - Z) U = \begin{bmatrix} 0 & -\widehat{R}_1^\top \Pi_n \\ \Pi_n \widehat{R}_1 & \Pi_n \widehat{T}_1 \Pi_n \end{bmatrix}, \tag{3.10b}$$

where $\Pi_n = [e_n, \ldots, e_1]$, $\widehat{R}_1 \in \mathbb{C}^{n\times n}$ is upper Hessenberg, $\widehat{R}_2, \widehat{R}_3, \widehat{R}_4 \in \mathbb{C}^{n\times n}$ are upper triangular, $\widehat{T}_1, \widehat{T}_2 \in \mathbb{C}^{n\times n}$ are skew symmetric, and $\widehat{T}_3 \in \mathbb{C}^{n\times n}$. Define

$$\mathcal{U}_0^\top := \left[\begin{array}{cc|cc} 0 & \Pi_n & 0 & 0 \\ 0 & 0 & 0 & \Pi_n \\ \hline 0 & 0 & I_n & 0 \\ -I_n & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} U^\top & 0 \\ 0 & V^\top \end{bmatrix}, \quad \mathcal{V}_0 := \mathcal{J}_{4n}^\top \mathcal{U}_0 \mathcal{J}_{4n}. \tag{3.11}$$

Then it is easily seen that $\mathcal{U}_0^H \widetilde{\mathcal{K}} \mathcal{V}_0$ and $\mathcal{U}_0^H \widetilde{\mathcal{N}} \mathcal{V}_0$ have the same forms as in (3.7) with "\hat" being over all submatrices. Furthermore, if we define

$$\mathcal{U}_b^\top = \begin{bmatrix} \mathcal{P}_{2n} & 0 \\ 0 & \mathcal{P}_{2n} \end{bmatrix} \mathcal{U}_0^\top, \quad \mathcal{V}_b := \mathcal{J}_{4n}^\top \mathcal{U}_b \mathcal{J}_{4n}, \tag{3.12}$$

then we have

$$\mathcal{U}_b^H \widetilde{\mathcal{K}} \mathcal{V}_b = \begin{bmatrix} K_{11}^b & K_{12}^b \\ 0 & (K_{11}^b)^\top \end{bmatrix}, \quad \mathcal{U}_b^H \widetilde{\mathcal{N}} \mathcal{V}_b = \begin{bmatrix} N_{11}^b & N_{12}^b \\ 0 & (N_{11}^b)^\top \end{bmatrix}, \tag{3.13}$$

where $K_{11}^b, K_{12}^b, N_{11}^b$, and N_{12}^b are of the same forms as in (3.4).

THEOREM 3.1. *If K_{11}^a and K_{11}^b are unreduced, and N_{11}^a and N_{11}^b are nonsingular (see* (3.4) *and* (3.13)*), then the SA_II is mathematically equivalent to the URV-based structured method.*

Proof. Denote $\mathcal{V}_a := [\mathcal{V}_1^a, \mathcal{V}_2^a]$ with $\mathcal{V}_i^a \in \mathbb{C}^{4n\times 2n} (i = 1, 2)$. Since $\mathcal{U}_a = \mathcal{J}_{4n}^\top \mathcal{V}_a \mathcal{J}_{4n}$, it holds that $\mathcal{U}_a = [\mathcal{J}_{4n}\mathcal{V}_2^a, -\mathcal{J}_{4n}\mathcal{V}_1^a]$. From (3.4), it follows that

$$\widetilde{\mathcal{K}} \mathcal{V}_1^a = \mathcal{J}_{4n} \mathcal{V}_2^a K_{11}^a, \quad \widetilde{\mathcal{N}} \mathcal{V}_1^a = \mathcal{J}_{4n} \mathcal{V}_2^a N_{11}^a. \tag{3.14}$$

This implies that

$$\widetilde{\mathcal{K}} \mathcal{V}_1^a = \widetilde{\mathcal{N}} \mathcal{V}_1^a (N_{11}^a)^{-1} K_{11}^a. \tag{3.15}$$

Since the first columns of $\mathcal{V}_1^a$ and $\mathcal{V}_1^b$ ($\mathcal{V}_b \equiv \left[\mathcal{V}_1^b, \mathcal{V}_2^b\right]$) are both e_1, by applying the implicit Q-theorem to (3.15), the matrices $\mathcal{U}_a$ and $\mathcal{V}_a$ are uniquely determined, and $\mathcal{U}_a = \mathcal{U}_b$ and $\mathcal{V}_a = \mathcal{V}_b$. ∎

4. $\top$-skew-Hamiltonian Arnoldi method. Based on SHIRA [15], in this section we briefly introduce the structure-preserving $\top$-skew-Hamiltonian Arnoldi algorithm to compute the desired eigenpairs of a $\top$-skew-Hamiltonian $\mathcal{B}$.

As in (2.4), using the $(\mathcal{S} + \mathcal{S}^{-1})$-transform, we transform $\mathcal{M} - \lambda\mathcal{L}$ of (1.3) into a $\top$-skew-Hamiltonian pencil $\mathcal{K} - \mu\mathcal{N}$ by

$$\mathcal{K} - \mu\mathcal{N} \equiv \left[(\mathcal{L}\mathcal{J}\mathcal{M}^\top + \mathcal{M}\mathcal{J}\mathcal{L}^\top) - \mu\mathcal{L}\mathcal{J}\mathcal{L}^\top\right] \mathcal{J}^\top. \tag{4.1}$$

Next, we derive the shift-invert transformation of $\mathcal{K} - \mu\mathcal{N}$. Let $\lambda_0 \notin \sigma(\mathcal{M}, \mathcal{L})$. Then, from Theorem 2.2(i), we have $\mu_0 \equiv \lambda_0 + \frac{1}{\lambda_0} \notin \sigma(\mathcal{K}, \mathcal{N})$. Define the shift-invert transformation $\widehat{\mathcal{K}} - \widehat{\mu}\widehat{\mathcal{N}}$ for $\mathcal{K} - \mu\mathcal{N}$ with $\widehat{\mu} = \frac{1}{\mu - \mu_0}$ and

$$\widehat{\mathcal{K}} \equiv -\lambda_0\mathcal{N} = -\lambda_0\mathcal{L}\mathcal{J}\mathcal{L}^\top\mathcal{J}^\top = \lambda_0 \begin{bmatrix} A_1^\top & 0 \\ 0 & A_1 \end{bmatrix}, \tag{4.2a}$$

$$\widehat{\mathcal{N}} \equiv -\lambda_0(\mathcal{K} - \mu_0\mathcal{N}) = -\lambda_0(\mathcal{L}\mathcal{J}\mathcal{M}^\top + \mathcal{M}\mathcal{J}\mathcal{L}^\top - \mu_0\mathcal{L}\mathcal{J}\mathcal{L}^\top)\mathcal{J}^\top. \tag{4.2b}$$

Substituting $\mu_0 = \lambda_0 + \frac{1}{\lambda_0}$ into (4.2b), $\widehat{\mathcal{N}}$ can be factorized as

$$\begin{aligned} \widehat{\mathcal{N}} &= -\lambda_0 \left(\mathcal{L}\mathcal{J}\mathcal{M}^\top + \mathcal{M}\mathcal{J}\mathcal{L}^\top - \left(\lambda_0 + \frac{1}{\lambda_0}\right) \mathcal{L}\mathcal{J}\mathcal{L}^\top\right) \mathcal{J}^\top \\ &= (\mathcal{M} - \lambda_0\mathcal{L})\, \mathcal{J} \left(\mathcal{M}^\top - \lambda_0\mathcal{L}^\top\right) \mathcal{J}^\top \equiv \mathcal{N}_1\mathcal{N}_2, \end{aligned} \tag{4.3}$$

where

$$\mathcal{N}_1 = \mathcal{M} - \lambda_0\mathcal{L}, \quad \mathcal{N}_2 = \mathcal{J}(\mathcal{M}^\top - \lambda_0\mathcal{L}^\top)\mathcal{J}^\top \tag{4.4}$$

are nonsingular and satisfy $\mathcal{N}_2^\top \mathcal{J} = \mathcal{J}\mathcal{N}_1$. The generalized eigenvalue problem $\widehat{\mathcal{K}} z = \widehat{\mu}\widehat{\mathcal{N}} z$ is then equivalent to the eigenvalue problem $\mathcal{B}y = \widehat{\mu} y$, where $y = \mathcal{N}_2 z$ and

$$\mathcal{B} \equiv \mathcal{N}_1^{-1} \widehat{\mathcal{K}} \mathcal{N}_2^{-1}. \tag{4.5}$$

Using the facts that $\widehat{\mathcal{K}}\mathcal{J} = \mathcal{J}\widehat{\mathcal{K}}^\top$ and $\mathcal{N}_2^\top \mathcal{J} = \mathcal{J}\mathcal{N}_1$, we find that $\mathcal{B}$ satisfies

$$\mathcal{J}\mathcal{B}^\top = \mathcal{J}\mathcal{N}_2^{-\top}\widehat{\mathcal{K}}^\top\mathcal{N}_1^{-\top} = \mathcal{N}_1^{-1}\mathcal{J}\widehat{\mathcal{K}}^\top\mathcal{N}_1^{-\top} = \mathcal{N}_1^{-1}\widehat{\mathcal{K}}\mathcal{J}\mathcal{N}_1^{-\top} = \mathcal{N}_1^{-1}\widehat{\mathcal{K}}\mathcal{N}_2^{-1}\mathcal{J} = \mathcal{B}\mathcal{J},$$

and hence $\mathcal{B}$ is again $\top$-skew-Hamiltonian.

We now define the Krylov matrix with respect to u_1 and j $(1 \le j \le n)$ by

$$K_j \equiv K_j[\mathcal{B}, u_1] = [u_1, \mathcal{B}u_1, \ldots, \mathcal{B}^{j-1}u_1] \tag{4.6}$$

and state two useful theorems from [15]. Note that these theorems are slightly different from the originals, but the proofs are almost identical to the ones in [15].

THEOREM 4.1 (see [15]). *Let $\mathcal{B} \in \mathbb{C}^{2n\times 2n}$ be $\top$-skew-Hamiltonian and $K_j \equiv K_j[\mathcal{B}, u_1]$ $(1 \le j \le n)$ be a Krylov matrix with rank$(K_j) = j$. Then span(K_j) is $\top$-isotropic and if $K_j = U_j\widehat{R}_j$ is a QR-factorization, then*

$$\mathcal{B}U_j = U_j\widehat{H}_j + \widehat{u}_{j+1}e_j^\top, \tag{4.7}$$

where $\widehat{H}_j \in \mathbb{C}^{j\times j}$ is unreduced upper Hessenberg, $U_j \in \mathbb{C}^{2n\times j}$ is orthonormal and $\top$-isotropic, and $\widehat{u}_{j+1} \in \mathbb{C}^{2n}$ is a suitable vector such that

$$U_j^H\widehat{u}_{j+1} = 0 \quad \textit{and} \quad U_j^\top\mathcal{J}\widehat{u}_{j+1} = 0. \tag{4.8}$$

THEOREM 4.2 (see [15]). *Let $\mathcal{B} \in \mathbb{C}^{2n\times 2n}$ be $\top$-skew-Hamiltonian. If rank $(K_n[\mathcal{B}, u_1]) = n$, then there is a unitary $\top$-symplectic matrix $\mathcal{U}$ with $\mathcal{U}e_1 = u_1$ such that*

$$\mathcal{U}^H\mathcal{B}\mathcal{U} = \begin{bmatrix} \widehat{H}_n & \widehat{N}_n \\ 0 & \widehat{H}_n^\top \end{bmatrix}, \tag{4.9}$$

where $\widehat{H}_n$ is unreduced upper Hessenberg and $\widehat{N}_n$ is $\top$-skew-symmetric.

Based on Theorem 4.2, the jth step of the Arnoldi process is given by

$$\widehat{h}_{j+1,j}u_{j+1} = \mathcal{B}u_j - \sum_{i=1}^{j}\widehat{h}_{ij}u_i, \tag{4.10}$$

where $\widehat{h}_{ij} = u_i^H\mathcal{B}u_j$, $i = 1, \ldots, j$, and $\widehat{h}_{j+1,j} > 0$ is chosen so that $\|u_{j+1}\|_2 = 1$. In order to ensure that the space span$\{u_1, \ldots, u_{j+1}\}$ is $\top$-isotropic to working precision, the jth step of the $\top$-isotropic Arnoldi process is modified by

$$\widehat{h}_{j+1,j}u_{j+1} = \mathcal{B}u_j - \sum_{i=1}^{j}\widehat{h}_{ij}u_i - \sum_{i=1}^{j}\widehat{t}_{ij}\mathcal{J}\bar{u}_i, \tag{4.11}$$

where $\widehat{h}_{ij} = u_i^H\mathcal{B}u_j$, $\widehat{t}_{ij} = -u_i^\top\mathcal{J}\mathcal{B}u_j$, $i = 1, \ldots, j$, and $\widehat{h}_{j+1,j} > 0$ is chosen so that $\|u_{j+1}\|_2 = 1$. We present the $\top$SHIRA-method.

ALGORITHM 4.1 ($\top$SHIRA).

Input: $\top$*-skew-Hamiltonian matrix* $\mathcal{B}$ *with starting vector* u_1.
Output: U_ℓ *and upper Hessenberg matrix* $\widehat{H}_\ell$ *with* $\mathcal{B}U_\ell = U_\ell\widehat{H}_\ell$, $U_\ell^H U_\ell = I_\ell$ *and* $U_\ell^\top \mathcal{J} U_\ell = 0$.
Use (4.11) *with starting vector* u_1 *to generate the* ℓ*th step of the* $\top$*-isotropic Arnoldi factorization:*

$$\mathcal{B}U_\ell = U_\ell\widehat{H}_\ell + \widehat{h}_{\ell+1,\ell}u_{\ell+1}e_\ell^\top.$$

For $k = 1, 2, \ldots,$ *until wanted* ℓ *eigenpairs of* $\mathcal{B}$ *are convergent,*
Use (4.11) *to extend the* ℓ*th step of the* $\top$*-isotropic Arnoldi factorization to the* $(\ell+p)$*th step of the* $\top$*-isotropic Arnoldi factorization:*

$$\mathcal{B}U_{\ell+p} = U_{\ell+p}\widehat{H}_{\ell+p} + \widehat{h}_{\ell+p+1,\ell+p}u_{\ell+p+1}e_{\ell+p}^\top.$$

Use standard implicitly restarted step for the Arnoldi algorithm [8] *to reform a new* ℓ*th step of the* $\top$*-isotropic Arnoldi factorization.*
End

Remark 4.1.

(i) $\widehat{h}_{\ell+1,\ell}$ is set to zero if $|\widehat{h}_{\ell+1,\ell}| < tol(|\widehat{h}_{\ell,\ell}| + |\widehat{h}_{\ell+1,\ell+1}|)$ for some stopping tolerance "tol."

(ii) Let (θ_i, v_i) be an eigenpair of $\widehat{H}_\ell$, i.e., $\widehat{H}_\ell v_i = \theta_i v_i$. Let $y_i = U_\ell v_i$ be the Ritz vector of $\mathcal{B}$ corresponding to the Ritz value θ_i. Then from (4.7) and (4.8), we have

$$\begin{aligned}
&\|\mathcal{B}y_i - \theta_i y_i\|_2 = \|\mathcal{B}U_\ell v_i - \theta_i U_\ell v_i\|_2 \\
&= \|(U_\ell\widehat{H}_\ell + \widehat{u}_{\ell+1,\ell}e_\ell^\top)v_i - \theta_i U_\ell v_i\|_2 \\
&= \|U_\ell(\widehat{H}_\ell v_i - \theta_i v_i) + \widehat{h}_{\ell+1,\ell}(e_\ell^\top v_i)u_{\ell+1}\|_2 \\
&= |\widehat{h}_{\ell+1,\ell}||e_\ell^\top v_i|.
\end{aligned}$$

5. Generalized $\top$-skew-Hamiltonian Arnoldi method. We now consider the generalized eigenvalue problem $\widehat{\mathcal{K}}z = \widehat{\mu}\widehat{\mathcal{N}}z$, where $\widehat{\mathcal{K}}$ and $\widehat{\mathcal{N}}$ are $\top$-skew-Hamiltonian given in (4.2). Based on the reduction method [16], $\widehat{\mathcal{K}} - \widehat{\mu}\widehat{\mathcal{N}}$ can be reduced to block triangular condensed forms

$$\mathcal{V}^\top(\widehat{\mathcal{K}} - \widehat{\mu}\widehat{\mathcal{N}})\mathcal{U} = \begin{bmatrix} K_{11} & K_{12} \\ 0 & K_{11}^\top \end{bmatrix} - \widehat{\mu}\begin{bmatrix} N_{11} & N_{12} \\ 0 & N_{11}^\top \end{bmatrix}, \tag{5.1}$$

where $K_{11}, N_{11} \in \mathbb{C}^{n\times n}$ are, respectively, upper Hessenberg and upper triangular, and $\mathcal{V}$ and $\mathcal{U} \in \mathbb{C}^{2n\times 2n}$ are unitary satisfying

$$\mathcal{V} = \mathcal{J}^\top\mathcal{U}\mathcal{J}. \tag{5.2}$$

In order to solve a large sparse product or a periodic eigenvalue problem, recently, a product (or a periodic) Arnoldi process and a product Krylov process were, respectively, proposed by Kressner's book [6, section 4.2.5] and Watkins' book [24, section 9.10]. Using the result of Theorem 4.1, we adopt the idea of the periodic Arnoldi process [6, section 4.2.5] to develop a generalized $\top$-skew-Hamiltonian algorithm which preserves the structure of (5.1) for the computation of the desired eigenpairs of $\widehat{\mathcal{K}}z = \widehat{\mu}\widehat{\mathcal{N}}z$.

THEOREM 5.1. *Let* $\mathcal{B} \equiv \mathcal{N}_1^{-1}\widehat{\mathcal{K}}\mathcal{N}_2^{-1}$ *be* $\top$*-skew-Hamiltonian defined in* (4.5). *Let* $\widehat{\mathcal{N}} = \mathcal{N}_1\mathcal{N}_2$ *and* $K_j \equiv K_j[\mathcal{B}, u_1]$ *be the Krylov matrix with rank*$(K_j) = j$. *If*

$$\mathcal{N}_2^{-1}K_j = Z_j R_{2,j} \quad \text{and} \quad \mathcal{N}_1 K_j = Y_j R_{1,j} \tag{5.3}$$

are QR-factorizations, where $Z_j, Y_j \in \mathbb{C}^{2n\times j}$ are orthonormal and $R_{2,j}, R_{1,j}$ are nonsingular upper triangular, then we have

$$\widehat{\mathcal{K}} Z_j = Y_j H_j + \widehat{y}_{j+1} e_j^\top \tag{5.4}$$

and

$$\widehat{\mathcal{N}} Z_j = Y_j R_j, \tag{5.5}$$

where $H_j \in \mathbb{C}^{j\times j}$ is unreduced upper Hessenberg, $R_j \in \mathbb{C}^{j\times j}$ is nonsingular upper triangular, and Y_j and Z_j are $\top$-bi-isotropic such that

$$Y_j^H \widehat{y}_{j+1} = 0 \quad \text{and} \quad Z_j^\top \mathcal{J} \widehat{y}_{j+1} = 0 \tag{5.6}$$

for a suitable $\widehat{y}_{j+1} \in \mathbb{C}^{2n}$.

Proof. Let $K_j = U_j \widehat{R}_j$ be the QR-factorization of K_j with $\widehat{R}_j$ being nonsingular upper triangular. From Theorem 4.1, it follows that

$$\mathcal{N}_1^{-1} \widehat{\mathcal{K}} \mathcal{N}_2^{-1} U_j = U_j \widehat{H}_j + \widehat{u}_{j+1} e_j^\top. \tag{5.7}$$

Substituting (5.3) into (5.7) we obtain

$$\begin{aligned}
\widehat{\mathcal{K}} Z_j &= \widehat{\mathcal{K}} \mathcal{N}_2^{-1} K_j R_{2,j}^{-1} = \widehat{\mathcal{K}} \mathcal{N}_2^{-1} U_j \widehat{R}_j R_{2,j}^{-1} \\
&= (\mathcal{N}_1 U_j \widehat{H}_j + \mathcal{N}_1 \widehat{u}_{j+1} e_j^\top) \widehat{R}_j R_{2,j}^{-1} \\
&= Y_j (R_{1,j} \widehat{R}_j^{-1} \widehat{H}_j \widehat{R}_j R_{2,j}^{-1}) + \gamma_j Y_j Y_j^H \mathcal{N}_1 \widehat{u}_{j+1} e_j^\top + \gamma_j (I - Y_j Y_j^H) \mathcal{N}_1 \widehat{u}_{j+1} e_j^\top \\
&= Y_j H_j + \widehat{y}_{j+1} e_j^\top, \qquad (5.8)
\end{aligned}$$

where $\gamma_j = e_j^\top \widehat{R}_j R_{2,j}^{-1} e_j$,

$$H_j = R_{1,j} \widehat{R}_j^{-1} \widehat{H}_j \widehat{R}_j R_{2,j}^{-1} + \gamma_j Y_j^H \mathcal{N}_1 \widehat{u}_{j+1} e_j^\top, \tag{5.9}$$

and

$$\widehat{y}_{j+1} = \gamma_j (I - Y_j Y_j^H) \mathcal{N}_1 \widehat{u}_{j+1}. \tag{5.10}$$

Since $\widehat{R}_j$, $R_{1,j}$, and $R_{2,j}$ are nonsingular upper triangular, and $\widehat{H}_j$ is unreduced upper Hessenberg, from (5.9) it follows that H_j is unreduced upper Hessenberg. Clearly, it holds that $Y_j^H \widehat{y}_{j+1} = 0$ by (5.10).

On the other hand, from (5.3), we also have

$$\widehat{\mathcal{N}} Z_j = \mathcal{N}_1 \mathcal{N}_2 Z_j = \mathcal{N}_1 K_j R_{2,j}^{-1} = Y_j R_{1,j} R_{2,j}^{-1} \equiv Y_j R_j,$$

where $R_j = R_{1,j} R_{2,j}^{-1}$ is nonsingular and upper triangular.

We now show that Y_j and Z_j are $\top$-bi-isotropic. By the fact that $\mathcal{N}_2^\top \mathcal{J} = \mathcal{J} \mathcal{N}_1$ and (5.3), it holds that

$$Y_j^\top \mathcal{J} Z_j = R_{1,j}^{-\top} K_j^\top (\mathcal{N}_1^\top \mathcal{J} \mathcal{N}_2^{-1}) K_j R_{2,j}^{-1} = R_{1,j}^{-\top} K_j^\top \mathcal{J} K_j R_{2,j}^{-1} = 0. \tag{5.11}$$

From (5.8) and (5.10), we have

$$Z_j^\top \mathcal{J} \widehat{y}_{j+1} e_j^\top = Z_j^\top \mathcal{J} (\widehat{\mathcal{K}} Z_j - Y_j H_j) = Z_j^\top \mathcal{J} \widehat{\mathcal{K}} Z_j,$$

which is $\top$-skew-symmetric. This implies that $Z_j^\top \mathcal{J} \widehat{y}_{j+1} = 0$. ∎

THEOREM 5.2. *Let* $\mathcal{B} = \mathcal{N}_1^{-1}\widehat{\mathcal{K}}\mathcal{N}_2^{-1}$ *be* $\top$*-skew-Hamiltonian defined in* (4.5) *and* $\widehat{\mathcal{N}} = \mathcal{N}_1\mathcal{N}_2$. *If rank*$(K_n[\mathcal{B}, u_1]) = n$, *then there are unitary matrices* $\mathcal{U}$ *and* $\mathcal{V}$ *satisfying* (5.2) *and* $\mathcal{V}e_1 = \mathcal{N}_1u_1/\|\mathcal{N}_1u_1\|_2$ *such that*

$$\mathcal{V}^\top\widehat{\mathcal{K}}\mathcal{U} = \begin{bmatrix} H_n & S_n \\ 0 & H_n^\top \end{bmatrix}, \quad \mathcal{V}^\top\widehat{\mathcal{N}}\mathcal{U} = \begin{bmatrix} R_n & T_n \\ 0 & R_n^\top \end{bmatrix}, \tag{5.12}$$

where H_n *is unreduced upper Hessenberg,* R_n *is nonsingular upper triangular, and* S_n *and* T_n *are* $\top$*-skew-symmetric.*

Proof. Applying Theorem 5.1 for $j = n$, we have $\widehat{y}_{n+1}$ being orthogonal to Y_n and $\mathcal{J}\bar{Z}_n$. This implies that $\widehat{y}_{n+1} = 0$. Then (5.4) and (5.5) become

$$\widehat{\mathcal{K}}Z_n = Y_nH_n \quad \text{and} \quad \widehat{\mathcal{N}}Z_n = Y_nR_n, \tag{5.13}$$

where H_n is unreduced upper Hessenberg and R_n is nonsingular upper triangular. Let $\mathcal{U} \equiv \begin{bmatrix} Z_n & -\mathcal{J}\bar{Y}_n \end{bmatrix}$, $\mathcal{V} \equiv \begin{bmatrix} Y_n & -\mathcal{J}\bar{Z}_n \end{bmatrix}$. Clearly,

$$Z_n^HZ_n = I_n, \quad Y_n^HY_n = I_n, \quad \text{and} \quad Y_n^\top\mathcal{J}Z_n = 0_n. \tag{5.14}$$

Then $\mathcal{U}$ and $\mathcal{V}$ satisfy (5.2). Since $\widehat{\mathcal{K}}\mathcal{J}$ and $\widehat{\mathcal{N}}\mathcal{J}$ are $\top$-skew symmetric, from (5.13)–(5.14), (5.12) follows. ∎

Based on Theorem 5.2, we now introduce a generalized $\top$-isotropic Arnoldi process which produces $\top$-bi-isotropic matrices Z_j and Y_{j+1} at the jth step.

By the recursive definition of j, let us first assume that the $\top$-bi-isotropic matrices Z_{j-1} and Y_j satisfy (5.4) and (5.5) with $j := j - 1$. That is, the $(j-1)$th step of the generalized $\top$-isotropic Arnoldi process generates

$$\widehat{\mathcal{N}}Z_{j-1} = Y_{j-1}R_{j-1}. \tag{5.15}$$

Now, we compare the jth columns of both sides in (5.5) which give

$$\widehat{\mathcal{N}}z_j = \sum_{i=1}^{j-1} r_{ij}y_i + r_{jj}y_j. \tag{5.16}$$

With (5.15), (5.16) it can be rewritten as

$$r_{jj}^{-1}z_j = \widehat{\mathcal{N}}^{-1}y_j - \sum_{i=1}^{j-1}\widehat{r}_{ij}z_i, \tag{5.17}$$

where

$$[\widehat{r}_{1j}, \ldots, \widehat{r}_{j-1,j}]^\top := -r_{jj}^{-1}R_{j-1}^{-1}[r_{1j}, \ldots, r_{j-1,j}]^\top. \tag{5.18}$$

Since $Z_j^HZ_j = I_j$, the coefficient $\widehat{r}_{ij}$ in (5.17) can be evaluated by

$$\widehat{r}_{ij} = z_j^H\widehat{\mathcal{N}}^{-1}y_j, \quad i = 1, \ldots, j-1, \tag{5.19}$$

and r_{jj} in (5.17) is chosen so that $\|z_j\|_2 = 1$. Substituting $[\widehat{r}_{1j}, \ldots, \widehat{r}_{j-1,j}]^\top$ of (5.19) into (5.18), we obtain the coefficient vector $[r_{1j}, \ldots, r_{j-1,j}]^\top$.

In exact arithmetic, z_j is orthogonal to $\mathcal{J}\bar{Y}_j$ automatically. As before, round-off errors cause $z_j^\top \mathcal{J} y_i$, $i = 1, \ldots, j$, to be tiny values. Thus, the jth step of the generalized $\top$-isotropic Arnoldi process for z_j should be modified by

$$r_{jj}^{-1} z_j = \widehat{\mathcal{N}}^{-1} y_j - \sum_{i=1}^{j-1} \widehat{r}_{ij} z_i - \sum_{i=1}^{j} s_{ij} \mathcal{J} \bar{y}_i, \tag{5.20a}$$

where

$$s_{ij} = y_i^\top \mathcal{J}^\top \left(\widehat{\mathcal{N}}^{-1} y_j - \sum_{i=1}^{j-1} \widehat{r}_{ij} z_i \right), \quad i = 1, \ldots, j. \tag{5.20b}$$

From (5.4), similar to (4.11), the jth step of the generalized $\top$-isotropic Arnoldi process for y_{j+1} is given by

$$h_{j+1,j} y_{j+1} = \widehat{\mathcal{K}} z_j - \sum_{i=1}^{j} h_{ij} y_i - \sum_{i=1}^{j} t_{ij} \mathcal{J} \bar{z}_i, \tag{5.21a}$$

where

$$h_{ij} = y_i^H \widehat{\mathcal{K}} z_j, \quad t_{ij} = z_i^\top \mathcal{J}^\top \widehat{\mathcal{K}} z_j, \quad i = 1, \ldots, j, \tag{5.21b}$$

and $h_{j+1,j} > 0$ is chosen so that $\|y_{j+1}\|_2 = 1$. Combing (5.20) and (5.21), we state the jth step of the generalized $\top$-isotropic Arnoldi process.

ALGORITHM 5.1 (the jth generalized $\top$-isotropic Arnoldi step).

Input: $\top$*-skew-Hamiltonian* $\widehat{\mathcal{K}}$ *and* $\widehat{\mathcal{N}}$*, upper triangular* $R(1:j-1, 1:j-1)$*,* $Y_j = [y_1, \ldots, y_j]$ *and* $Z_{j-1} = [z_1, \ldots, z_{j-1}]$ *with* $Y_j^H Y_j = I_j$*,* $Z_{j-1}^H Z_{j-1} = I_{j-1}$*, and* $Y_j^\top \mathcal{J} Z_{j-1} = 0$*.*

Output: $[h_{1,j}, \ldots, h_{j+1,j}]$*,* $R(1:j, 1:j)$*,* y_{j+1}*, and* z_j*.*

Compute z_j *in* (5.20) *by using the modified Gram–Schmidt step:*

Solve $\widehat{\mathcal{N}} z_j = y_j$*;*

For $i = 1, \ldots, j-1$

$\widehat{r}_{ij} = z_i^H z_j$, $z_j = z_j - \widehat{r}_{ij} z_i$

End

Set $R(j,j) := \|z_j\|_2^{-1}$*,* $z_j := R(j,j) z_j$*, and*

$R(1:j-1, j) := -R(j,j) R(1:j-1, 1:j-1)[\widehat{r}_{1j}, \cdots, \widehat{r}_{j-1,j}]^\top$*;*

Reorthogonalize z_j *to* $\mathcal{J}\bar{Y}_j$*:*

For $i = 1, \ldots, j$

$s_{ij} = y_i^\top \mathcal{J}^\top z_j$, $z_j = z_j - s_{ij} \mathcal{J} \bar{y}_i$

End

Compute y_{j+1} *in* (5.21)*:*

Compute $y_{j+1} = \widehat{\mathcal{K}} z_j$*;*

For $i = 1, \ldots, j$

$h_{ij} = y_i^H y_{j+1}$, $y_{j+1} = y_{j+1} - h_{ij} y_i$

End

Set $h_{j+1,j} := \|y_{j+1}\|_2$ *and* $y_{j+1} := y_{j+1} / h_{j+1,j} y_{j+1}$*;*

For $i = 1, \ldots, j$

$t_{ij} = z_i^\top \mathcal{J}^\top y_{j+1}$, $y_{j+1} = y_{j+1} - t_{ij} \mathcal{J} \bar{z}_i$

End

5.1. Implicitly restart. We now derive the implicitly restarted step for the $(\ell+p)$th step of the generalized $\top$-isotropic Arnoldi process. Suppose we have computed the $(\ell+p)$th step of the generalized $\top$-isotropic Arnoldi factorization:

$$\widehat{\mathcal{K}} Z_{\ell+p} = Y_{\ell+p} H_{\ell+p} + h_{\ell+p+1,\ell+p} y_{\ell+p+1} e_{\ell+p}^\top, \tag{5.22}$$

$$\widehat{\mathcal{N}} Z_{\ell+p} = Y_{\ell+p} R_{\ell+p}. \tag{5.23}$$

Let $\{\lambda_1, \ldots, \lambda_\ell, \lambda_{\ell+1}, \ldots, \lambda_{\ell+p}\}$ be the eigenvalues of the matrix pair $(H_{\ell+p}, R_{\ell+p})$, where $\{\lambda_1, \ldots, \lambda_\ell\}$ are the wanted eigenvalues. Let Q_k and V_k for $k = 1, \ldots, p$ be unitary matrices computed by the implicit-QZ step [22, p. 147] for $(H_{\ell+p}, R_{\ell+p})$ with the single shift $\lambda_{\ell+k}$.

Let $\widehat{H}_{\ell+p} := Q_p^H \cdots Q_1^H H_{\ell+p} V_1 \cdots V_p$, $\widehat{R}_{\ell+p} := Q_p^H \cdots Q_1^H R_{\ell+p} V_1 \cdots V_p$, $\widehat{Y}_{\ell+p} := Y_{\ell+p} Q_1 \cdots Q_p$, and $\widehat{Z}_{\ell+p} := Z_{\ell+p} V_1 \cdots V_p$. Then $\widehat{H}_{\ell+p}$ and $\widehat{R}_{\ell+p}$ are upper Hessenberg and upper triangular, respectively, and $\widehat{Y}_{\ell+p}$ and $\widehat{Z}_{\ell+p}$ satisfy $\widehat{Y}_{\ell+p}^\top \mathcal{J} \widehat{Z}_{\ell+p} = 0$ because of $Y_{\ell+p}^\top \mathcal{J} Z_{\ell+p} = 0$. Multiplying (5.22) and (5.23) by $V_1 \cdots V_p$, we get

$$\widehat{\mathcal{K}} \widehat{Z}_{\ell+p} = \widehat{Y}_{\ell+p} \widehat{H}_{\ell+p} + h_{\ell+p+1,\ell+p} y_{\ell+p+1} e_{\ell+p}^\top V_1 \cdots V_p, \tag{5.24}$$

$$\widehat{\mathcal{N}} \widehat{Z}_{\ell+p} = \widehat{Y}_{\ell+p} \widehat{R}_{\ell+p}. \tag{5.25}$$

Since

$$e_{\ell+p}^\top V_1 = \alpha_{\ell+p} e_{\ell+p-1}^\top + \beta_{\ell+p} e_{\ell+p}^\top,$$

by induction, the first $\ell - 1$ entries of $e_{\ell+p}^\top V_1 \cdots V_p$ are zero. Hence a new ℓth step of the generalized $\top$-isotropic Arnoldi factorization can be obtained by equating the first ℓ columns of (5.24) and (5.25):

$$\widehat{\mathcal{K}} \widehat{Z}_\ell = \widehat{Y}_\ell \widehat{H}_\ell + \widehat{h}_{\ell+p+1,\ell+p} y_{\ell+p+1} e_\ell^\top,$$
$$\widehat{\mathcal{N}} \widehat{Z}_\ell = \widehat{Y}_\ell \widehat{R}_\ell.$$

We summarize the above processes in Algorithm 5.2.

ALGORITHM 5.2 (generalized implicitly restarted step).

Input: given $(Y_{\ell+p}, y_{\ell+p+1}, Z_{\ell+p}, H_{\ell+p}, h_{\ell+p+1,\ell+p}, R_{\ell+p})$;
Output: $(Y_\ell, y_{\ell+1}, Z_\ell, H_\ell, h_{\ell+1,\ell}, R_\ell)$ formed a new ℓth step of the generalized $\top$-isotropic Arnoldi factorization. The best ℓ eigenvalues are locked in (H_ℓ, R_ℓ).
  Sort the eigenvalues of $(H_{\ell+p}, R_{\ell+p})$ from best to worst according to the sorting criterion and take $\{\lambda_{\ell+1}, \ldots, \lambda_{\ell+p}\}$ to be the p worst eigenvalues.
  Set $v := h_{\ell+p+1,\ell+p} e_{\ell+p}$;
  For $k = 1, \ldots, p$,
    Compute unitary matrices Q_k and V_k by the implicit-QZ step for $(H_{\ell+p}, R_{\ell+p})$ with the single shift $\lambda_{\ell+k}$ so that $Q_k^H H_{\ell+p} V_k$ and $Q_k^H R_{\ell+p} V_k$ are upper Hessenberg and upper triangular, respectively;
    Update $Y_{\ell+p} := Y_{\ell+p} Q_k$, $Z_{\ell+p} := Z_{\ell+p} V_k$, $H_{\ell+p} := Q_k^H H_{\ell+p} V_k$, $R_{\ell+p} := Q_k^H R_{\ell+p} V_k$, $v := Z_k^H v$;
  End
  Set $H_\ell = H_{\ell+p}(1:\ell, 1:\ell)$, $h_{\ell+1,\ell} := e_\ell^\top v$, $R_\ell = R_{\ell+p}(1:\ell, 1:\ell)$, $Y_\ell := Y_{\ell+p}(:, 1:\ell)$, $y_{\ell+1} := y_{\ell+p+1}$, $Z_\ell := Z_{\ell+p}(:, 1:\ell)$.

We now present the G⊤SHIRA.

ALGORITHM 5.3 (G⊤SHIRA).

Input: $\top$*-skew-Hamiltonian matrices* $\widehat{\mathcal{K}}$ *and* $\widehat{\mathcal{N}}$ *with starting vector* y_1.
Output: Z_ℓ, Y_ℓ, *upper Hessenberg* H_ℓ, *and upper triangular* R_ℓ *with*
$\widehat{\mathcal{K}}Z_\ell = Y_\ell H_\ell$, $\widehat{\mathcal{N}}Z_\ell = Y_\ell R_\ell, Y_\ell^H Y_\ell = I_\ell$, $Z_\ell^H Z_\ell = I_\ell$, *and* $Y_\ell^\top \mathcal{J} Z_\ell = 0$.
Use Algorithm 5.1 *with starting vector* y_1 *to generate an* ℓ*th step of the generalized* $\top$*-isotropic Arnoldi factorization:*

$$\widehat{\mathcal{K}}Z_\ell = Y_\ell H_\ell + h_{\ell+1,\ell} y_{\ell+1} e_\ell^\top,$$
$$\widehat{\mathcal{N}}Z_\ell = Y_\ell R_\ell.$$

For $k = 1, 2, \ldots,$ *until wanted* ℓ *eigenpairs of* $(\widehat{\mathcal{K}}, \widehat{\mathcal{N}})$ *are convergent,*
Use Algorithm 5.1 *to extend the* ℓ*th step of the generalized* $\top$*-isotropic Arnoldi factorization to the* $(\ell+p)$*th step of the generalized* $\top$*-isotropic Arnoldi factorization:*

$$\widehat{\mathcal{K}}Z_{\ell+p} = Y_{\ell+p} H_{\ell+p} + h_{\ell+p+1,\ell+p} y_{\ell+p+1} e_{\ell+p}^\top,$$
$$\widehat{\mathcal{N}}Z_{\ell+p} = Y_{\ell+p} R_{\ell+p}.$$

Use Algorithm 5.2 *to reform a new* ℓ*th step of the generalized* $\top$*-isotropic Arnoldi factorization.*
End

Remark 5.1.

(i) $h_{\ell+1,\ell}$ is set to zero if $|h_{\ell+1,\ell}| < tol(|h_{\ell,\ell}| + |h_{\ell+1,\ell+1}|)$ for some stopping tolerance "*tol*."

(ii) Let (θ_i, v_i) be an eigenpair of (H_ℓ, R_ℓ), i.e., $H_\ell v_i = \theta_i R_\ell v_i$, and let $z_i = Z_\ell v_i$ be a Ritz vector of the eigenproblem $\widehat{\mathcal{K}}z = \mu\widehat{\mathcal{N}}z$ corresponding to the Ritz value θ_i. Then from (5.4) and (5.5), we have

$$\begin{aligned}
&\|\widehat{\mathcal{K}}z_i - \theta_i\widehat{\mathcal{N}}z_i\|_2 = \|\widehat{\mathcal{K}}Z_\ell v_i - \theta_i\widehat{\mathcal{N}}Z_\ell v_i\|_2 \\
&= \|(Y_\ell H_\ell + h_{\ell+1,\ell} y_{\ell+1} e_\ell^\top)v_i - \theta_i Y_\ell R_\ell v_i\|_2 \\
&= \|Y_\ell(H_\ell v_i - \theta_i R_\ell v_i) + h_{\ell+1,\ell}(e_\ell^\top v_i) y_{\ell+1}\|_2 \\
&= \|h_{\ell+1,\ell}(e_\ell^\top v_i) y_{\ell+1}\|_2 = |h_{\ell+1,\ell}||e_\ell^\top v_i|.
\end{aligned}$$

6. Numerical study: Vibration of fast trains. In this section, we shall study the resonance phenomena of a railway track under high frequent excitation forces. We present numerical results of the vibration of fast trains to illustrate the performance of the proposed structure-preserving algorithms in sections 2–5. All numerical experiments are carried out using MATLAB 2006b with the machine precision $eps \approx 2.22 \times 10^{-16}$.

Research in the vibration of fast trains contributes to the safety of operations of high-speed trains as well as new designs of train bridges, embedded rail structures (ERS), and train suspension systems. Recently, the dynamic response of the vehicle-rail-bridge interaction system under different train speed was studied in [25] and a procedure for designing an optimal ERS was proposed in [14]. In both papers, the accurate numerical estimation to the resonance frequencies of the rail plays an important role. However, as mentioned by Ipsen in [5], the classical finite element packages fail to deliver correct resonance frequency for such problems. In this section, we would like to use our structure-preserving algorithms to solve the palindromic QEP (1.1) arising from the spectral modal analysis of rails under periodic excitation forces.

In the model of vibration of fast trains, we assume that the rail sections between

consecutive sleeper bays are identical, the distance between consecutive wheels is the same, and the wheel loads are equal. The rail between two sleepers is modeled by a three-dimensional isotropic elastic solid with linear isoparametric tetrahedron finite elements. Figure 6.1 shows a three-dimensional rail model (see [1] for details).

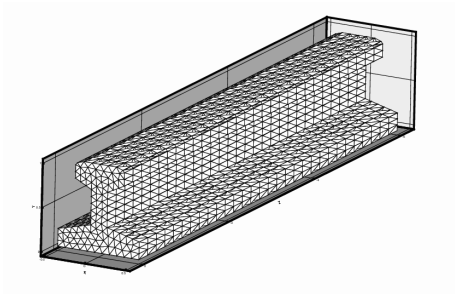


FIG. 6.1. *A three-dimensional rail model.*

Based on the ERS design [14], the external force is assumed to be periodic and the displacements of two boundary cross sections of the modeled rail are assumed to have a ratio λ, which is dependent on the excitation frequency of the external force. From the virtual work principle and strain-stress relationship, the governing equation for the displacement vector q involving viscous damping can be formulated by $Kq + D\dot{q} + M\ddot{q} = f(t)$, where K, D, and M from the finite element discretization on a uniform mesh satisfy the given boundary conditions. These matrices have the form

$$\begin{bmatrix} E_{11} & \widetilde{E}_{1,2:m-1}^{\top} & \frac{1}{\lambda}E_{m,m+1} \\ \widetilde{E}_{1,2:m-1} & \widetilde{E}_{2:m-1} & \widetilde{E}_{2:m-1,m}^{\top} \\ \lambda E_{m,m+1}^{\top} & \widetilde{E}_{2:m-1,m} & E_{m,m} \end{bmatrix}$$

in which $\widetilde{E}_{1,2:m-1}^{\top} = [E_{12}^{\top}, 0_n, \ldots, 0_n]$, $\widetilde{E}_{2:m-1,m} = [0_n, \ldots, 0_n, E_{m-1,m}]$, and $\widetilde{E}_{2:m-1} = \operatorname{tridiag}\big(E_{i-1,i}, E_{i,i}, E_{i,i+1}^{\top}\big)_{i=2}^{m-1}$ with $E_{ij} \in \mathbb{R}^{n\times n}$, $i, j = 1, \ldots, m+1$. (See [1] for details.) Furthermore, from the spectral modal analysis, we consider $q = \widetilde{x}e^{i\omega t}$, where ω is the frequency of the external force and $\widetilde{x}$ is the corresponding eigenmode. Consequently, we get the palindromic QEP

$$\left(\lambda^2 \widetilde{A}_1^{\top} + \lambda \widetilde{A}_0 + \widetilde{A}_1\right)\widetilde{x} = 0, \tag{6.1}$$

where

$$\left[\widetilde{A}_1\right]_{ij} = \begin{cases} K_{m,m+1} + i\omega D_{m,m+1} - \omega^2 M_{m,m+1} & (\text{if } \ i = m, j = 1), \\ 0 & \text{otherwise}, \end{cases}$$

$$\left[\widetilde{A}_0\right]_{ij} = \begin{cases} K_{i,j} + i\omega D_{i,j} - \omega^2 M_{i,j} & (\text{if } \ i-1 \le j \le i+1), \\ 0 & \text{otherwise}. \end{cases}$$

By consulting the preprocessing procedure (see [4] or [1]) for the deflation of all trivial zero and infinite eigenvalues of (6.1), we arrive to the deflated palindromic QEP

$$\left(\lambda^2 A_1^{\top} + \lambda A_0 + A_1\right) x = 0. \tag{6.2}$$

Example 6.1. We first consider the deflated palindromic QEP (6.2) for high-speed trains and rails. The size of A_0 and A_1 after deflation is $n = 303$, and the excitation frequency ω is chosen as 1000. The absolute values of the eigenvalues vary from 10^{-20} to 10^{20}.

We compute all eigenpairs of Example 6.1 by the SA_I, SA_II, and QZ algorithm. Note that as shown in section 3, SA_II and the URV-based method [20] are mathematically equivalent. In practice, we compare the backward error (relative residual (RRes)) of (1.1) by SA_II and the SKURV software [18]. Since SKURV gives only the eigenvalues, the associated eigenvectors are computed from (3.9) and (3.10) by inverse iteration. Numerical results show that the backward errors obtained by SA_II and SKURV for Example 6.1 are slightly different. Therefore, in the following computation, we adapt SA_II instead of the URV-method.

To measure the accuracy of an approximate eigenpair (λ, x) for (6.2), we use the RRes

$$\text{RRes} \equiv \frac{\|\lambda^2 A_1^\top x + \lambda A_0 x + A_1 x\|_2}{(|\lambda^2|\|A_1\|_F + |\lambda|\|A_0\|_F + \|A_1\|_F)\|x\|_2}. \tag{6.3}$$

As mentioned before, theoretically, the eigenvalues of (6.2) appear in pairs $(\lambda, \frac{1}{\lambda})$. So, if we sort the eigenvalues in the ascending order by modulus, the product of the ith and $(2n + 1 - i)$th sorted eigenvalues should be one. Therefore, we define the reciprocities of computed eigenvalues by

$$|\lambda_i \lambda_{2n+1-i} - 1|, \ i = 1, \ldots, n. \tag{6.4}$$

The RRes of the computed eigenpairs by the SA_I, SA_II, and QZ algorithm for the eigenvalues with absolute values in $[10^{-20}, 10^{20}]$ and $\omega = 1000$ are shown in Figure 6.2. For eigenvalues with small modulus, the SA_I performs much better than the SA_II and the QZ algorithm. For eigenvalues near the unit circle or with large modulus, all three algorithms have similar accuracy.

The important reciprocity property of eigenvalues is shown in Figure 6.3. Clearly, SA_I and SA_II preserve the essential reciprocity property as expected, while the QZ algorithm has only less than 12 pairs of computed eigenvalues near the unit circle with reciprocity near zero ($\approx 1.17 \times 10^{-12}$). The average and maximal values of all reciprocities are 0.220 and 1.006, respectively.

Next, we apply the SA_I, SA_II, and QZ algorithm to the palindromic QEP with various excitation frequency ω. Figure 6.4 shows the RRes of all computed eigenpairs with eigenvalues in $[10^{-20}, 10^{20}]$ by the three algorithms for 100 different ω's uniformly chosen from 50 to 5000. We see that the RRes of the SA_I are better than those of the SA_II and the QZ algorithm for all ω's.

Example 6.2. *We now consider the palindromic QEP* (6.1) *for high-speed trains and rails, with n, the size of* A_0 *and* A_1*, being* 5757.

Computational cost. Before showing our numerical results computed by the $\top$SHIRA and G$\top$SHIRA, we compare the computational costs of one step of the $\top$-isotropic Arnoldi process and the implicitly restarted step in each algorithm.

In one step of the $\top$SHIRA, it requires one matrix-vector product for $\mathcal{B}$, and $3j$ inner products and saxpy operations with vector length $2n$. Since $\mathcal{B} = \mathcal{N}_1^{-1}\widetilde{\mathcal{K}}\mathcal{N}_2^{-1}$, by the definitions of $\widetilde{\mathcal{K}}$ and $(\mathcal{N}_1, \mathcal{N}_2)$ in (4.2a) and (4.4), the matrix-vector of $\mathcal{B}$ requires solving 2 linear systems, 4 and 2 matrix-vector products for A_1 and A_0, respectively, and 6 saxpy operations with vector length n.

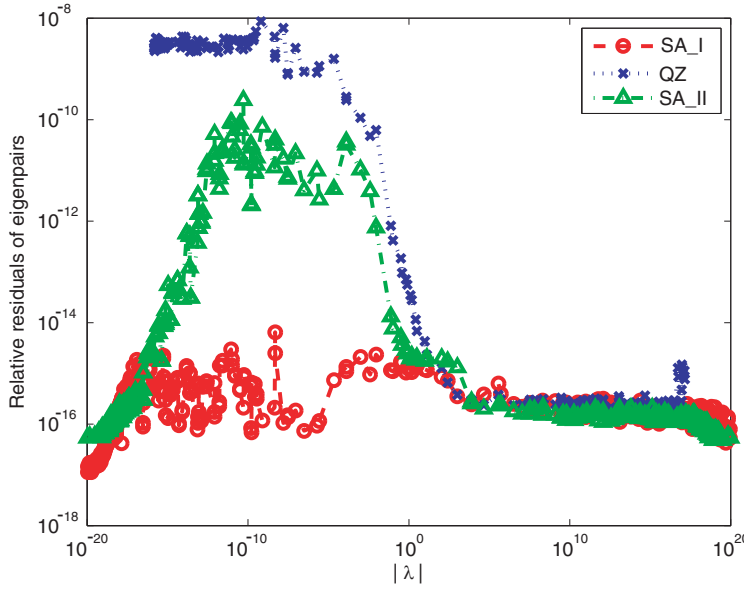


FIG. 6.2. *RRes of Example* 6.1 *($\omega = 1000$).*

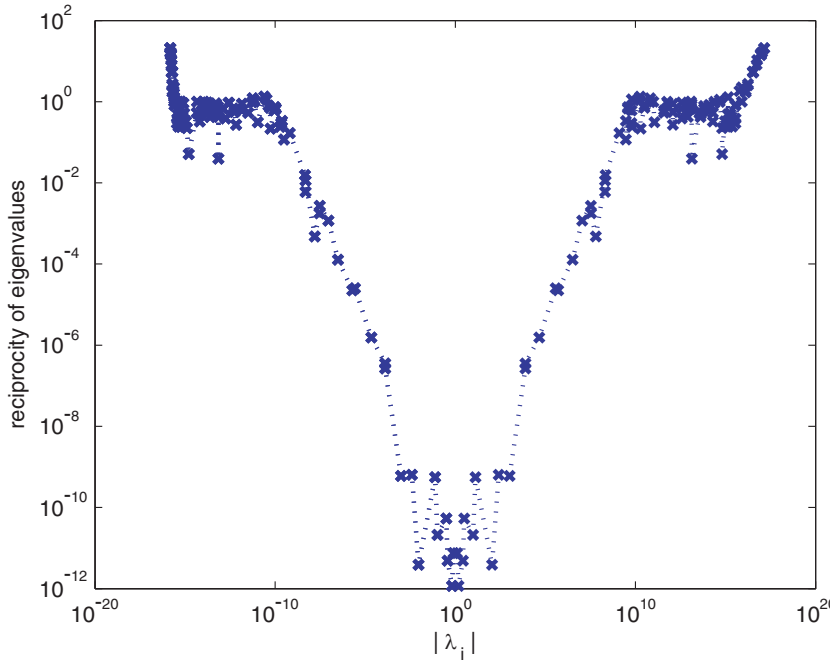


FIG. 6.3. *Reciprocities of computed eigenvalues produced by the QZ algorithm ($\omega = 1000$).*

In one step of the G⊤SHIRA, solving $\widetilde{z}_j$ requires solving 2 linear systems, 2 matrix-vector products of A_0 and A_1, and 6 saxpy operations with vector length n; computing z_j requires $2j - 1$ inner products and saxpy operations with vector length $2n$;

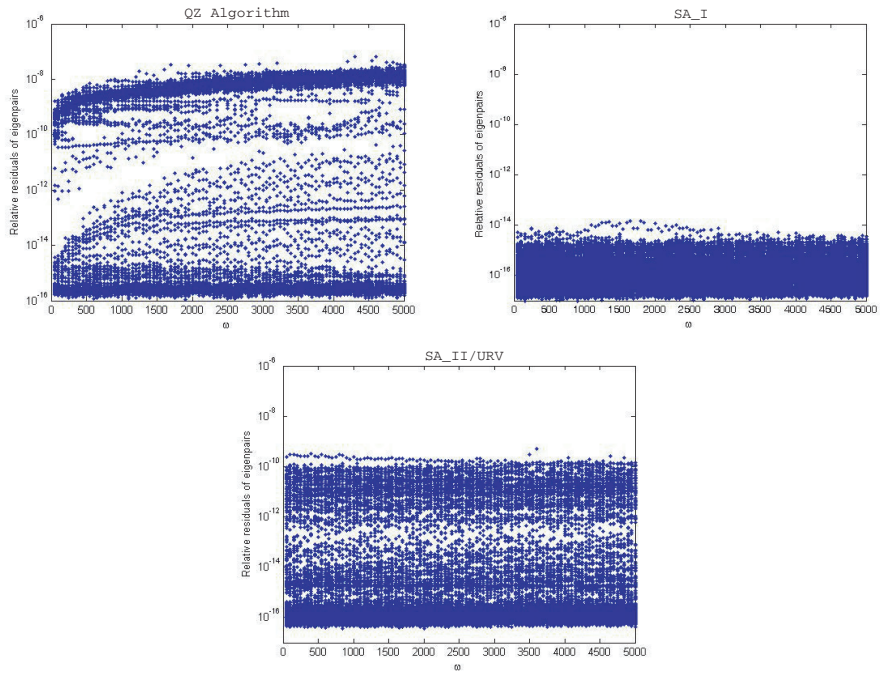


FIG. 6.4. *The RRes of eigenvalues vs. ω.*

TABLE 6.1
Computational cost of one step of the ⊤-isotropic Arnoldi process in the ⊤SHIRA and G⊤SHIRA algorithms.

	⊤SHIRA	G⊤SHIRA
Solving linear system	2	2
Matrix-vector product for A_1	4	4
Matrix-vector product for A_0	2	2
Inner products	$6j$	$8j-2$
Saxpy operations	$6j+6$	$8j+4$

computing y_{j+1} requires 2 matrix-vector products for A_1, and $2j$ inner products and saxpy operations with vector length $2n$.

We summarize the above computational costs in Table 6.1. The vector length of the inner products and saxpy operations in Table 6.1 is equal to n. On the other hand, the implicitly restarted steps in the ⊤SHIRA and G⊤SHIRA require $2(\ell+p-1)p$ and $4(\ell+p-1)p$ saxpy operations with vector length $2n$, respectively. Comparing one ⊤-isotropic Arnoldi step with one implicitly restarted step, the G⊤SHIRA algorithm is slightly more expensive than the ⊤SHIRA algorithm.

Accuracy of eigenpairs. We now compare the numerical results computed by the ⊤SHIRA and G⊤SHIRA algorithms. Here, $\lambda_{\omega,1},\ldots,\lambda_{\omega,10}$ denote target eigenvalues, and we set $\ell=10$, $p=20$ in the implicitly restarted step for each algorithm.

The RRes of $(\lambda_{\omega,i},x_i)$ and $(\frac{1}{\lambda_{\omega,i}},\widetilde{x}_i)$ for $i=1,\ldots,10$ are shown in Figure 6.5, where x_i and $\widetilde{x}_i$ are the corresponding computed eigenvectors. In (a) and (b) of Figure 6.5, we show those RRes for frequency $\omega=50$ and $\omega=2000$, respectively. The notations "Δ" and "$\times$" denote the results computed by the ⊤SHIRA and G⊤SHIRA

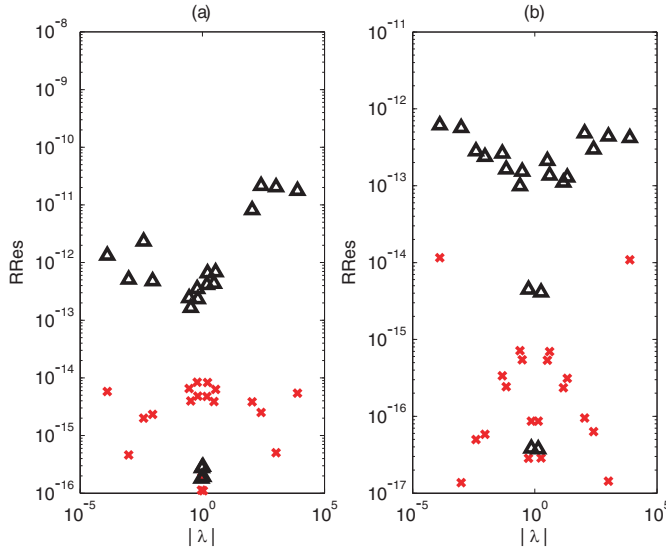


FIG. 6.5. *The RRes of the eigenpairs computed by the ⊤SHIRA and G⊤SHIRA algorithms. The notations "Δ" and "×" denote the results computed by the ⊤SHIRA and G⊤SHIRA algorithms, respectively. In* (a) *and* (b)*, the frequency ω is equal to* 50 *and* 2000*, respectively.*

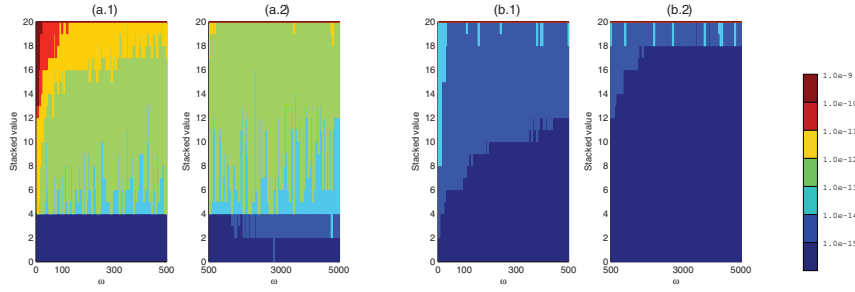


FIG. 6.6. *The stacked bars of $\ell_{\omega,k}$ for $k = 1, \ldots, 7$ with different ω. For each ω, all $\ell_{\omega,k}$ for $k = 1, \ldots, 7$ are stacked to form a vertical bar with ordering $\ell_{\omega,1}, \ell_{\omega,2}, \ldots, \ell_{\omega,7}$. Each bar is multicolored and the color corresponds to distinct $\ell_{\omega,k}$. The color bar in the right position shows the relationship between color and interval I_k, which corresponds to $\ell_{\omega,k}$. The results in* (a) *and* (b) *are computed by the ⊤SHIRA and G⊤SHIRA algorithms, respectively.*

algorithms, respectively. From these results, we see that the reciprocity property of the eigenvalues are preserved in both algorithms, but the accuracy of the eigenpairs computed by the G⊤SHIRA algorithm is obviously better than that by the ⊤SHIRA algorithm.

In order to give an overall comparison between the two algorithms, we compute the eigenpairs $(\lambda_{\omega,i}, x_i)$ and $(\frac{1}{\lambda_{\omega,i}}, \widetilde{x}_i)$ for $i = 1, \ldots, 10$ with $\omega = 5, 10, 15, \ldots, 500$ and $\omega = 550, 600, 650, \ldots, 5000$. We analyze the distribution of the corresponding 20 RRes with respect to ω. We partition the interval $(0, 10^{-9})$ into seven subintervals $\mathcal{I}_1 = (0, 10^{-15}]$, $\mathcal{I}_2 = (10^{-15}, 10^{-14}], \ldots, \mathcal{I}_7 = (10^{-10}, 10^{-9})$. For fixed ω, let $\ell_{\omega,k}$ be the number of the RRes which belongs to the interval $\mathcal{I}_k$ for $k = 1, \ldots, 7$. In Figure 6.6, for each ω, all $\ell_{\omega,k}$, $k = 1, \ldots, 7$, are stacked to form a vertical bar with ordering

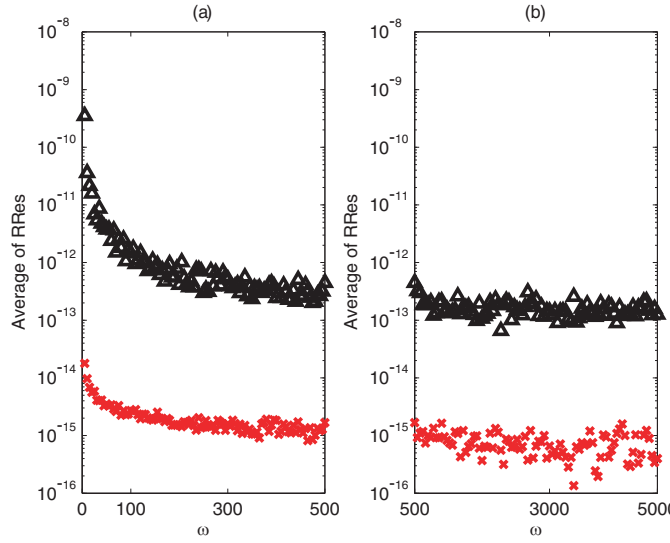


FIG. 6.7. *The average of RRes for the twelve eigenpairs computed by the* ⊤*SHIRA and G*⊤*SHIRA algorithms. The notations "*Δ*" and "*$\times$*" denote the results computed by the* ⊤*SHIRA and G*⊤*SHIRA algorithms, respectively.*

$\ell_{\omega,1}, \ell_{\omega,2}, \ldots, \ell_{\omega,7}$. The bar height is 20 which is the sum of $\ell_{\omega,1}, \ldots, \ell_{\omega,7}$. Each bar is multicolored and the color corresponds to distinct $\ell_{\omega,k}$. The color bar in the right bottom position of Figure 6.6 shows the relationship between colors and intervals $\mathcal{I}_k$ corresponding to $\ell_{\omega,k}$. All stacked bars of $\ell_{\omega,k}$ $(k = 1, \ldots, 7)$ with $\omega = 5, 10, 15, \ldots, 500$ are shown in (a.1) and (b.1) of Figure 6.6 and those with $\omega = 550, 600, 650, \ldots, 5000$ are shown in (a.2) and (b.2) of Figure 6.6. The results in (a) and (b) of Figure 6.6 are computed by the ⊤SHIRA and the G⊤SHIRA algorithms, respectively.

In the above paragraph, we show the distribution of the RRes for different ω for the comparison of the accuracy of the target eigenpairs. From another point of view, we show the average of the RRes for the target eigenpairs with each ω in Figure 6.7. The notations "Δ" and "$\times$" in Figure 6.7 denote the results computed by the ⊤SHIRA and G⊤SHIRA algorithms, respectively. From Figures 6.6 and 6.7, we can summarize that the accuracy of the eigenpairs computed by the G⊤SHIRA algorithm are obviously better than that of the ⊤SHIRA algorithm for all ω in $(0, 5000]$.

We now try to explain the different accuracies of the two algorithms. One important reason is that the ⊤SHIRA algorithm needs to solve a linear system in the extraction method of eigenvectors, while the G⊤SHIRA algorithm needs only vector additions. The accuracy of the extracted eigenvector will be reduced if the condition number of the linear system is large. On the other hand, Theorem A.1 in Appendix A.3 may help explain this phenomenon from the viewpoint of the minimal residual. The accuracy of the eigenpair computed by the G⊤SHIRA algorithm is better than that by the ⊤SHIRA algorithm, since the G⊤SHIRA algorithm is a generalized Arnoldi algorithm for $\widehat{\mathcal{K}}z = \widehat{\mu}\widehat{\mathcal{N}}z$, while the ⊤SHIRA algorithm is an Arnoldi algorithm for $\mathcal{N}_1^{-1}\widehat{\mathcal{K}}\mathcal{N}_2^{-1}y = \widehat{\mu}y$.

7. Conclusions. In this paper, we first transform a palindromic QEP to a ⊤-skew-Hamiltonian pencil by the $(S + S^{-1})$-transform. Then, we extend Patel's ap-

proach to solve the $\top$-skew-Hamiltonian pencil efficiently. We have also developed a structure-preserving generalized $\top$-skew-Hamiltonian implicitly restarted Arnoldi method (G$\top$SHIRA) for solving the large sparse $\top$-skew-Hamiltonian pencil. Numerical results show that the accuracy of desired eigenpairs computed by the G$\top$SHIRA is better than that computed by the classical $\top$SHIRA. The standard algorithms proposed in this paper are numerically stable for solving palindromic QEPs. In the future, we are motivated to develop structure-preserving algorithms for solving the antipalindromic QEP $\lambda^2 A_1^\top + \lambda A_0 - A_1$ with $A_0^\top = -A_0$, efficiently.

Appendix.

A.1. In this section we list pseudocodes of Step 2 in Algorithm 2.1.

In the following, $\mathsf{givensl}(\alpha,\beta,\mathsf{i})$ returns a Givens rotation G such that $G[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}] = \gamma e_i$ with $\gamma \in \mathbb{C}$; $\mathsf{givensr}(\alpha,\beta,\mathsf{i})$ returns a Givens rotation G such that $[\alpha\ \beta]G = \gamma e_i^\top$ with $\gamma \in \mathbb{C}$. The functions $\mathsf{qr}(\mathsf{A})$ and $\mathsf{ql}(\mathsf{A})$ perform the standard QR and QL factorizations.

Step 2 *in Algorithm* 2.1. function $[\mathcal{K},\mathcal{N},Q,Z] = \mathsf{rbutf}(\mathcal{K},\mathcal{N})$

Input: Matrices $\mathcal{K},\mathcal{N}$ in the form (2.4).

Output: Unitary Q, Z and $\mathcal{K},\mathcal{N}$ of the form (2.5), where $\mathcal{K}$ and $\mathcal{N}$ are overwritten by $Q\mathcal{K}Z$ and $Q\mathcal{N}Z$, respectively.

$[Q_1, R] \leftarrow \mathsf{qr}(\mathcal{N}(1:n, 1:n))$
$Q \leftarrow \mathrm{diag}(\mathrm{Q_1^H},\ \mathrm{I_n})$
$Z \leftarrow \mathrm{diag}(\mathrm{I_n},\ \bar{\mathrm{Q}}_1)$
$\mathcal{K} \leftarrow Q\mathcal{K}Z$
$\mathcal{N} \leftarrow Q\mathcal{N}Z$
for $j = 1 : n-2$
  for $k = j+1 : n-1$
    % *annihilate* $\mathcal{K}(n+k, j)$ *by Givens rotation in* $(n+k, n+k+1)$ *plane*
    $G \leftarrow \mathsf{givensl}(\mathcal{K}(n+k, j), \mathcal{K}(n+k+1, j), 2)$
    $Q(n+k : n+k+1, :) \leftarrow GQ(n+k : n+k+1, :)$
    $\mathcal{K}(n+k : n+k+1, :) \leftarrow G\mathcal{K}(n+k : n+k+1, :)$
    $\mathcal{N}(n+k : n+k+1, :) \leftarrow G\mathcal{N}(n+k : n+k+1, :)$
    $Z(:, k : k+1) \leftarrow Z(:, k : k+1)G^\top$
    $\mathcal{K}(:, k : k+1) \leftarrow \mathcal{K}(:, k : k+1)G^\top$
    $\mathcal{N}(:, k : k+1) \leftarrow \mathcal{N}(:, k : k+1)G^\top$
    % *annihilate* $\mathcal{N}(k+1, k)$ *by Givens rotation in* $(k, k+1)$ *plane*
    $G \leftarrow \mathsf{givensl}(\mathcal{N}(k, k), \mathcal{N}(k+1, k), 1)$
    $Q(k : k+1, :) \leftarrow GQ(k : k+1, :)$
    $\mathcal{K}(k : k+1, :) \leftarrow G\mathcal{K}(k : k+1, :)$
    $\mathcal{N}(k : k+1, :) \leftarrow G\mathcal{N}(k : k+1, :)$
    $Z(:, n+k : n+k+1) \leftarrow Z(:, n+k : n+k+1)G^\top$
    $\mathcal{K}(:, n+k : n+k+1) \leftarrow \mathcal{K}(:, n+k : n+k+1)G^\top$
    $\mathcal{N}(:, n+k : n+k+1) \leftarrow \mathcal{N}(:, n+k : n+k+1)G^\top$
  end
  % *annihilate* $\mathcal{N}(2n, j)$ *by Givens rotation in* $(n, 2n)$ *plane*
  $G \leftarrow \mathsf{givensl}(\mathcal{N}(n, j), \mathcal{N}(2n, j), 1)$
  $Q([n\ 2n], :) \leftarrow GQ([n\ 2n], :)$
  $\mathcal{K}([n\ 2n], :) \leftarrow G\mathcal{K}([n\ 2n], :)$
  $\mathcal{N}([n\ 2n], :) \leftarrow G\mathcal{N}([n\ 2n], :)$
  $Z(:, [n\ 2n]) \leftarrow Z(:, [n\ 2n])G^H$
  $\mathcal{K}(:, [n\ 2n]) \leftarrow \mathcal{K}(:, [n\ 2n])G^H$

32: $\mathcal{N}(:,[n\ 2n]) \leftarrow \mathcal{N}(:,[n\ 2n])G^H$
33: **for** $k = n:-1:j+2$
34: % *annihilate* $\mathcal{K}(k,j)$ *by Givens rotation in* $(k-1,k)$ *plane*
35: $G \leftarrow$ givensl$(\mathcal{K}(k-1,j), \mathcal{K}(k,j), 1)$
36: $Q(k-1:k,:) \leftarrow GQ(k-1:k,:)$
37: $\mathcal{K}(k-1:k,:) \leftarrow G\mathcal{K}(k-1:k,:)$
38: $\mathcal{N}(k-1:k,:) \leftarrow G\mathcal{N}(k-1:k,:)$
39: $Z(:,n+k-1:n+k) \leftarrow Z(:,n+k-1:n+k)G^\top$
40: $\mathcal{K}(:,n+k-1:n+k) \leftarrow \mathcal{K}(:,n+k-1:n+k)G^\top$
41: $\mathcal{N}(:,n+k-1:n+k) \leftarrow \mathcal{N}(:,n+k-1:n+k)G^\top$
42: % *annihilate* $\mathcal{N}(k,k-1)$ *by Givens rotation in* $(k-1,k)$ *plane*
43: $G \leftarrow$ givensr$(\mathcal{N}(k,k-1), \mathcal{N}(k,k), 2)$
44: $Q(n+k-1:n+k,:) \leftarrow G^\top Q(n+k-1:n+k,:)$
45: $\mathcal{K}(n+k-1:n+k,:) \leftarrow G^\top \mathcal{K}(n+k-1:n+k,:)$
46: $\mathcal{N}(n+k-1:n+k,:) \leftarrow G^\top \mathcal{N}(n+k-1:n+k,:)$
47: $Z(:,k-1:k) \leftarrow Z(:,k-1:k)G$
48: $\mathcal{K}(:,k-1:k) \leftarrow \mathcal{K}(:,k-1:k)G$
49: $\mathcal{N}(:,k-1:k) \leftarrow \mathcal{N}(:,k-1:k)G$
50: **end**
51: **end**

A.2. To show the extra zeros of the subdiagonals of the submatrices in (3.4), let $\mathbb{H}_k$ and $\mathbb{T}_k$ be the sets of $k\times k$ upper Hessenberg and triangular matrices, respectively, and let $\mathbb{S}_{2k}$ be the set of $2k\times 2k$ $\top$-skew symmetric matrices. Denote

$$\text{(A2.1)}\quad \mathbb{A}_{2k} = \left\{ A\in\mathbb{C}^{2k\times 2k} \middle| A \equiv P_{2k}^\top \left[\begin{array}{c|c} 0_k & \diagdown \\ \hline \diagup & 0_k \end{array}\right] P_{2k} \ \text{ with } \diagdown\in\mathbb{H}_k \ \text{ and } \diagup\in\mathbb{T}_k \right\},$$

where $P_{2k} = [e_1, e_{k+1}, e_2, e_{k+2}, \ldots, e_k, e_{2k}]$,

$$\text{(A2.2)}\qquad \mathbb{R}_{2k} = \left\{ R\in\mathbb{C}^{2k\times 2k} \middle| R \equiv P_{2k}^\top \left[\begin{array}{c|c} 0_k & \diagup \\ \hline \diagup & 0_k \end{array}\right] P_{2k} \ \text{ with } \diagup\in\mathbb{T}_k \right\},$$

$$\text{(A2.3)}\quad \mathbb{B}_{2m,2k} = \{B\in\mathbb{C}^{2m\times 2k} | Be_1 = Be_3 = \cdots = Be_{2k-1} = 0\},$$

$$\text{(A2.4)}\quad \widehat{\mathbb{B}}_{2m,2k} = \{\widehat{B}\in\mathbb{C}^{2m\times 2k} | \widehat{B}e_2 = \widehat{B}e_4 = \cdots = \widehat{B}e_{2k} = 0\},$$

$$\text{(A2.5)}\quad \mathbb{C}_{2m\times 2k} = \{C\in\mathbb{C}^{2m\times 2k} | c_{ij} = 0, i = 1,\ldots,2m, j = 1,\ldots,2k \ \text{ and } \ (i,j)\neq(1,2k)\},$$

$$\text{(A2.6)}\quad \mathbb{D}_{2k} = \{D\in\mathbb{C}^{2k\times 2k} | D\in\mathbb{S}_{2k} \ \text{ with } \ \{1,-1,3,-3,\ldots,2k-1,-(2k-1)\} \text{ – diagonals being zeros}\},$$

$$\text{(A2.7)}\quad \widehat{\mathbb{D}}_{2k} = \{\widehat{D}\in\mathbb{C}^{2k\times 2k} | \widehat{D}\in\mathbb{S}_{2k} \ \text{ with } \ \{2,-2,4,-4,\ldots,2k-2,-(2k-2)\} \text{ – diagonals being zeros}\}.$$

After performing the first and second steps of the SA_I (i.e., Steps 07–50 in section A.1, for $j = 1$ and 2) on $(\widetilde{\mathcal{K}}, \widetilde{\mathcal{N}})$, it produces

$$\text{(A2.8a)}\quad K_{11}^{(2)} := \left[\begin{array}{cc|ccc} 0 & \times & 0 & \cdots & 0 \\ \times & 0 & \times & \cdots & \times \\ \hline 0 & \times & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{array}\right],\quad K_{12}^{(2)} := \left[\begin{array}{cc|ccc} 0 & 0 & \times & \cdots & \times \\ 0 & 0 & 0 & \cdots & 0 \\ \hline \times & 0 & & & \\ \vdots & \vdots & & G_{2n-2} & \\ \times & 0 & & & \end{array}\right],$$

$$\text{(A2.8b)}\qquad K_{21}^{(2)} := \left[\begin{array}{cc|ccc} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & & & \\ \vdots & \vdots & & H_{2n-2} & \\ 0 & 0 & & & \end{array}\right],\ K_{22}^{(2)} := (K_{11}^{(2)})^{\top},$$

and

$$\text{(A2.9a)}\ N_{11}^{(2)} := \left[\begin{array}{cc|ccc} \times & 0 & \times & \cdots & \times \\ 0 & \times & 0 & \cdots & 0 \\ \hline 0 & 0 & & & \\ \vdots & \vdots & & T_{2n-2} & \\ 0 & 0 & & & \end{array}\right],\ N_{12}^{(2)} := \left[\begin{array}{cc|ccc} 0 & \times & 0 & \cdots & 0 \\ \times & 0 & \times & \cdots & \times \\ \hline 0 & \times & & & \\ \vdots & \vdots & & 0_{2n-2} & \\ 0 & \times & & & \end{array}\right],$$

$$\text{(A2.9b)}\qquad N_{21}^{(2)} := 0_{2n},\quad N_{22}^{(2)} := (N_{11}^{(2)})^{\top},$$

where G_{2n-2} and $H_{2n-2} \in \mathbb{S}_{2n-2}$ and $T_{2n-2} \in \mathbb{T}_{2n-2}$. Let $m = n - k$. Suppose after $2k$ steps (for $j = 1, 2, \ldots, 2k$) the SA_I gives

$$\text{(A2.10a)}\ K_{11}^{(2k)} := \begin{bmatrix} A_{2k} & B_{2m,2k}^{\top} \\ C_{2m,2k} & 0_{2m} \end{bmatrix},\quad K_{12}^{(2k)} := \begin{bmatrix} D_{2k} & \widehat{B}_{2m,2k}^{\top} \\ \widehat{B}_{2m,2k} & G_{2m} \end{bmatrix},$$

$$\text{(A2.10b)}\ K_{21}^{(2k)} := \begin{bmatrix} 0_{2k} & 0_{2m,2k}^{\top} \\ 0_{2m,2k} & H_{2m} \end{bmatrix},\quad K_{22}^{(2k)} := (K_{11}^{(2k)})^{\top},$$

and

$$\text{(A2.11a)}\quad N_{11}^{(2k)} := \begin{bmatrix} R_{2k} & \widehat{E}_{2m,2k}^{\top} \\ 0_{2m,2k} & T_{2m} \end{bmatrix},\ N_{12}^{(2k)} := \begin{bmatrix} \widehat{D}_{2k} & E_{2m,2k}^{\top} \\ E_{2m,2k} & 0_{2m} \end{bmatrix},$$

$$\text{(A2.11b)}\qquad N_{21}^{(2k)} := 0_{2n},\quad N_{22}^{(2k)} := (N_{11}^{(2k)})^{\top},$$

where $A_{2k} \in \mathbb{A}_{2k}$, $R_{2k} \in \mathbb{R}_{2k}$, $C_{2m,2k} \in \mathbb{C}_{2m,2k}$, $B_{2m,2k}$, $E_{2m,2k} \in \mathbb{B}_{2m,2k}$, $\widehat{B}_{2m,2k}$, $\widehat{E}_{2m,2k} \in \widehat{\mathbb{B}}_{2m,2k}$, $D_{2k} \in \mathbb{D}_{2k}$, $\widehat{D}_{2k} \in \widehat{\mathbb{D}}_{2k}$, $G_{2m}, H_{2m} \in \mathbb{S}_{2m}$, and $T_{2m} \in \mathbb{T}_{2m}$.

By letting $k' = k+1$ and $m' = m-1$, we perform the SA_I for $j = 2k+1, 2k+1$ and obtain

$$\text{(A2.12a)}\ K_{11}^{(2k')} := \begin{bmatrix} A_{2k'} & B_{2m',2k'}^{\top} \\ C_{2m',2k'} & 0_{2m'} \end{bmatrix},\quad K_{12}^{(2k')} := \begin{bmatrix} D_{2k'} & \widehat{B}_{2m',2k'}^{\top} \\ \widehat{B}_{2m',2k'} & G_{2m'} \end{bmatrix},$$

$$\text{(A2.12b)}\ K_{21}^{(2k')} := \begin{bmatrix} 0_{2k'} & 0_{2m',2k'}^{\top} \\ 0_{2m',2k'} & H_{2m'} \end{bmatrix},\quad K_{22}^{(2k')} := (K_{11}^{(2k')})^{\top},$$

and

$$\text{(A2.13a)}\ N_{11}^{(2k')} := \begin{bmatrix} R_{2k'} & \widehat{E}_{2m',2k'}^{\top} \\ 0_{2m',2k'} & T_{2m'} \end{bmatrix},\ N_{12}^{(2k')} := \begin{bmatrix} \widehat{D}_{2k'} & E_{2m',2k'}^{\top} \\ E_{2m',2k'} & 0_{2m'} \end{bmatrix},$$

$$\text{(A2.13b)}\qquad N_{21}^{(2k')} := 0_{2n},\quad N_{22}^{(2k')} := (N_{11}^{(2k')})^{\top},$$

where the subblocks in (A2.12)–(A2.13) have the same forms as in (A2.10)–(A2.11) by replacing k and m by k' and m', respectively, and satisfy

$$A_{2k} = \Phi_{2k}^{\top} A_{2k'} \Phi_{2k}, \quad D_{2k} = \Phi_{2k}^{\top} D_{2k} \Phi_{2k}, \tag{A2.14a}$$

$$R_{2k} = \Phi_{2k}^{\top} R_{2k'} \Phi_{2k}, \quad \widehat{D}_{2k} = \Phi_{2k}^{\top} \widehat{D}_{2k'} \Phi_{2k}, \tag{A2.14b}$$

where $\Phi_{2k} = [e_1, \ldots, e_{2k}]$ with $e_i \in \mathbb{C}^{2k'}, i = 1, \ldots, 2k$. By the inductive process above, (3.4) holds with $k' = n$ in (A2.12)–(A2.13) and with the superscript "a" in (3.4) being $(2n)$. ∎

A.3. THEOREM A.1. *Let $V \in \mathbb{C}^{n\times r}$ be a unitary matrix and $A, B \in \mathbb{C}^{n\times n}$. Then*

$$\|AV - BVC\|_2 \geq \|AV - BVP\|_2 \qquad \text{for all } C \in \mathbb{C}^{r\times r},$$

where $P = (V^H B^H BV)^{-1}(V^H B^H AV)$, or equivalently, $P = (U^H BV)^{-1}(U^H AV)$, where $BV = US$ is the QR factorization of BV.

Proof. Since

$$\begin{aligned} -C^H V^H B^H BVP &= -C^H V^H B^H BV(V^H B^H BV)^{-1}(V^H B^H AV) \\ &= -C^H V^H B^H AV, \end{aligned}$$

it follows that

$$\begin{aligned} &(V^H A^H - C^H V^H B^H)(AV - BVC) \\ &= V^H A^H AV - C^H V^H B^H AV - V^H A^H BVC + C^H V^H B^H BVC \\ &= V^H A^H AV + (P^H - C^H)V^H B^H BV(P - C) - P^H V^H B^H BVP \\ &= (V^H A^H - P^H V^H B^H)(AV - BVP) + (P^H - C^H)V^H B^H BV(P - C). \end{aligned}$$

Obviously, $(P^H - C^H)V^H B^H BV(P - C)$ is semidefinite. Then by Weyl's theorem, we have

$$\lambda_j((AV - BVC)^H(AV - BVC)) \geq \lambda_j((AV - BVP)^H(AV - BVP)),\ j = 1, \ldots, n.$$

Hence

$$\|AV - BVC\|_2 \geq \|AV - BVP\|_2,$$

since $\|G\|_2^2 = \lambda\max(G^H G)$. ∎

REFERENCES

[1] E. K.-W. CHU, T.-M. HWANG, W.-W. LIN, AND C.-T. WU, *Vibration of fast trains, palindromic eigenvalue problems and structure-preserving doubling algorithms*, J. Comput. Appl. Math., 219 (2008), pp. 237–252.

[2] J. J. HENCH AND A. J. LAUB, *Numerical solution of the discrete-time periodic Riccati equation*, IEEE Trans. Automat. Control, 39 (1994), pp. 1197–1210.

[3] A. HILLIGES, *Numerische Lösung von quadratischen eigenwertproblemen mit Anwendungen in der Schiendynamik*, Master's thesis, Technical University Berlin, Berlin, Germany, 2004.

[4] A. HILLIGES, C. MEHL, AND V. MEHRMANN, *On the solution of palindromic eigenvalue problems*, in Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS), Jyväskylä, Finland, 2004.

[5] I. C. F. IPSEN, *Accurate eigenvalues for fast trains*, SIAM News, 37, SIAM, Philadelphia, 2004.

[6] D. Kressner, *Numerical methods for general and structured eigenvalue problems*, Lect. Notes Comput. Sci. Eng. 46, Springer, Berlin, 2005.
[7] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.
[8] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.
[9] W.-W. Lin, *A new method for computing the closed-loop eigenvalues of a discrete-time algebraic Riccati equation*, Linear Algebra Appl., 96 (1987), pp. 157–180.
[10] W.-W. Lin and S.-F. Xu, *Convergence analysis of structure-preserving doubling algorithms for Riccati-type matrix equations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 26–39.
[11] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Numerical methods for palindromic eigenvalue problems: Computing the anti-triangular Schur form*, Numer. Linear Algebra Appl., to appear.
[12] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Structured polynomial eigenvalue problems: Good vibrations from good linearizations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1029–1051.
[13] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 971–1004.
[14] V. L. Markine, A. P. de Man, S. Jovanovic, and C. Esveld, *Optimal design of embedded rail structure for high-speed railway lines*, in Railway Engineering 2000, 3rd International Conference, London, 2000.
[15] V. Mehrmann and D. Watkins, *Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils*, SIAM J. Sci. Comput., 22 (2001), pp. 1905–1925.
[16] R. V. Patel, *On computing the eigenvalues of a symplectic pencil*, Linear Algebra Appl., 188 (1993), pp. 591–611.
[17] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.
[18] C. Schröder, *SKURV: A Matlab Toolbox for the Skew URV Decomposition of a Matrix Triple*, http://www.math.tu-berlin.de/∼schroed/Software/skurv.
[19] C. Schröder, *A QR-like Algorithm for the Palindromic Eigenvalue Problem*, Technical report, Preprint 388, TU Berlin, Berlin, Germany, 2007.
[20] C. Schröder, *URV decomposition based structured methods for palindromic and even eigenvalue problems*, Technical report, Preprint 375, TU Berlin, Berlin, Germany, 2007.
[21] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
[22] G. W. Stewart, *Matrix Algorithms, Volume* II: *Eigensystems*, SIAM, Philadelphia, 2001.
[23] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.
[24] D. S. Watkins, *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*, SIAM, Philadelphia, 2007.
[25] Y. S. Wu, Y. B. Yang, and J. D. Yau, *Three-dimensional analysis of train-rail-bridge interaction problems*, Vehicle Syst. Dyn., 35 (2001), pp. 1–35.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1593–1613

DETECTING AND SOLVING HYPERBOLIC QUADRATIC EIGENVALUE PROBLEMS*

CHUN-HUA GUO†, NICHOLAS J. HIGHAM‡, AND FRANÇOISE TISSEUR‡

Abstract. Hyperbolic quadratic matrix polynomials $Q(\lambda) = \lambda^2 A + \lambda B + C$ are an important class of Hermitian matrix polynomials with real eigenvalues, among which the overdamped quadratics are those with nonpositive eigenvalues. Neither the definition of overdamped nor any of the standard characterizations provides an efficient way to test if a given Q has this property. We show that a quadratically convergent matrix iteration based on cyclic reduction, previously studied by Guo and Lancaster, provides necessary and sufficient conditions for Q to be overdamped. For weakly overdamped Q the iteration is shown to be generically linearly convergent with constant at worst 1/2, which implies that the convergence of the iteration is reasonably fast in almost all cases of practical interest. We show that the matrix iteration can be implemented in such a way that when overdamping is detected a scalar $\mu < 0$ is provided that lies in the gap between the n largest and n smallest eigenvalues of the $n \times n$ quadratic eigenvalue problem (QEP) $Q(\lambda)x = 0$. Once such a μ is known, the QEP can be solved by linearizing to a definite pencil that can be reduced, using already available Cholesky factorizations, to a standard Hermitian eigenproblem. By incorporating an initial preprocessing stage that shifts a hyperbolic Q so that it is overdamped, we obtain an efficient algorithm that identifies and solves a hyperbolic or overdamped QEP maintaining symmetry throughout and guaranteeing real computed eigenvalues.

Key words. quadratic eigenvalue problem, hyperbolic, overdamped, weakly overdamped, quadratic matrix polynomial, quadratic matrix equation, solvent, cyclic reduction, doubling algorithm

AMS subject classifications. 15A18, 15A24, 65F15, 65F30

DOI. 10.1137/070704058

1. Introduction.

The quadratic eigenvalue problem (QEP) is to find scalars λ and nonzero vectors x and y satisfying $Q(\lambda)x = 0$ and $y^*Q(\lambda) = 0$, where

$$Q(\lambda) = \lambda^2 A + \lambda B + C, \quad A, B, C \in \mathbb{C}^{n\times n} \tag{1.1}$$

is a quadratic matrix polynomial. The vectors x and y are right and left eigenvectors, respectively, corresponding to the eigenvalue λ. The many applications of the QEP, as well as its theory and algorithms for solving it, are surveyed by Tisseur and Meerbergen [27].

Our interest in this work is in Hermitian quadratic matrix polynomials—those with Hermitian A, B, and C—and more specifically those that are *hyperbolic*. Hyperbolic quadratics, and the subclass of overdamped quadratics, are defined as follows. For Hermitian X and Y we write $X > Y$ ($X \geq Y$) if $X - Y$ is positive definite (positive semidefinite).

*Received by the editors October 1, 2007; accepted for publication (in revised form) by Q. Ye September 15, 2008; published electronically January 16, 2009.

http://www.siam.org/journals/simax/30-4/70405.html

†Department of Mathematics and Statistics, University of Regina, Regina, SK S4S 0A2, Canada (chguo@math.uregina.ca, http://www.math.uregina.ca/~chguo/). The research of this author was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.

‡School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK (higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/, ftisseur@ma.man.ac.uk, http://www.ma.man.ac.uk/~ftisseur/). The work of both authors was supported by Engineering and Physical Sciences Research Council grant EP/D079403. The second author's research was also supported by a Royal Society-Wolfson Research Merit Award.

DEFINITION 1.1. *$Q(\lambda)$ is hyperbolic if A, B, and C are Hermitian, $A > 0$, and*

$$(x^*Bx)^2 > 4(x^*Ax)(x^*Cx) \quad \textit{for all nonzero } x \in \mathbb{C}^n. \tag{1.2}$$

DEFINITION 1.2. *$Q(\lambda)$ is overdamped if it is hyperbolic with $B > 0$ and $C \geq 0$.*

Overdamped quadratics arise in overdamped systems in structural mechanics [22, section 7.6].

Any eigenpair of Q satisfies $x^*Q(\lambda)x = 0$ and hence

$$\lambda = \frac{-x^*Bx \pm \sqrt{(x^*Bx)^2 - 4(x^*Ax)(x^*Cx)}}{2x^*Ax}. \tag{1.3}$$

Therefore the eigenvalues of a hyperbolic Q are real and those of an overdamped Q are real and nonpositive. Both classes of quadratics have other important spectral properties, which we summarize in section 2.

We have two aims. The first is to devise an efficient and reliable numerical test for hyperbolicity or overdamping of a given Hermitian quadratic. The second aim is to build upon an affirmative test result an efficient algorithm for solving the QEP that exploits hyperbolicity and in particular that guarantees real computed eigenvalues in floating point arithmetic.

Part of the motivation for testing overdamping concerns the stability of gyroscopic systems. It is known that a gyroscopic system $G(\lambda) = \lambda^2 A_g + \lambda B_g + C_g$ with $A_g, C_g > 0$ and B_g Hermitian indefinite and nonsingular is stable whenever the quadratic $Q_g(\lambda) = \lambda^2 A_g + \lambda|B_g| + C_g$ is overdamped [9]. Here $|B_g|$ is the Hermitian positive definite square root of B_g^2 (i.e., the Hermitian polar factor of the Hermitian matrix B_g) [12].

Checking the hyperbolicity condition (1.2) is a nontrivial task, and plausible sufficient conditions for hyperbolicity may be incorrect. For example, it is claimed in [21] that when $A = I$, $B > 0$, and $C \geq 0$, Q is hyperbolic if $B > 2C^{1/2}$. That this claim is false has been shown by Barkwell and Lancaster [1].

Guo and Lancaster [9] propose testing overdamping by using a matrix iteration based on cyclic reduction to compute two solvents (solutions) of the quadratic matrix equation

$$Q(X) = AX^2 + BX + C = 0 \tag{1.4}$$

and then computing an extremal eigenvalue of each solvent. A definiteness test on Q evaluated at the average of the two extremal eigenvalues finally determines whether Q is overdamped. We show that the same iteration can be used to test overdamping in a much more efficient way that does not necessarily require the iteration to be run to convergence, even for a positive test result. Our test is based on a more complete understanding of the behavior of the matrix iteration, developed in section 3.

In section 4 we extend the convergence analysis to weakly overdamped quadratics, for which the strict inequality in (1.2) is replaced by a weak inequality ($\geq$). The key idea is to use an interpretation of the matrix iteration as a doubling algorithm. We show that for weakly overdamped Q with equality in (1.2) for some nonzero x, the iteration is linearly convergent with constant at worst $1/2$ in the generic case. A reasonable speed of convergence can therefore be expected in almost all practically important cases.

In section 5 we turn to algorithmic matters. We first show how a hyperbolic Q can be shifted to make it overdamped. Then we specify our test for overdamping, which

requires only the building blocks of Cholesky factorization, matrix multiplication, and the solution of triangular systems. We then show how after a successful test the eigensystem of an overdamped Q can be efficiently computed in a way that exploits the symmetry and definiteness and guarantees real computed eigenvalues.

Veselić [28] and Higham, Tisseur, and Van Dooren [19] have previously shown that every hyperbolic quadratic can be reformulated as a definite pencil $L(\lambda) = \lambda X + Y$ of twice the dimension, and this connection is explored in detail and in more generality by Higham, Mackey, and Tisseur [16]. However, the algorithm developed here is the first practical procedure for arranging that X or Y is a definite matrix and hence allowing symmetry and definiteness to be fully exploited.

Section 6 concludes the paper with a numerical experiment that provides further insight into the theory and algorithms.

2. Preliminaries. We first recall the definition of a definite pencil.

DEFINITION 2.1. *A Hermitian pencil $L(\lambda) = \lambda X + Y$ is definite (or, equivalently, the matrices X, Y form a definite pair) if $(z^*Xz)^2 + (z^*Yz)^2 > 0$ for all nonzero $z \in \mathbb{C}^n$.*

Definite pairs have the desirable properties that they are simultaneously diagonalizable under congruence and, in the associated eigenproblem $L(\lambda)x = 0$, the eigenvalues are real and semisimple.[1]

The next result gives three conditions each equivalent to the condition (1.2) in the definition of hyperbolic quadratic.

THEOREM 2.2. *Let the $n \times n$ quadratic $Q(\lambda) = \lambda^2 A + \lambda B + C$ be Hermitian with $A > 0$ and let*

$$\gamma = \min_{\|x\|_2=1} [(x^*Bx)^2 - 4(x^*Ax)(x^*Cx)]. \tag{2.1}$$

The following statements are equivalent:

(a) *Q is hyperbolic.*

(b) *$\gamma > 0$.*

(c) *$x^*Q(\lambda)x = 0$ has two distinct real zeros for all nonzero $x \in \mathbb{C}^n$.*

(d) *$Q(\mu) < 0$ for some $\mu \in \mathbb{R}$.*

Proof. (a) $\Leftrightarrow$ (b) $\Leftrightarrow$ (c) is immediate. (c) $\Leftrightarrow$ (d) follows from Markus [25, Lemma 31.15]. ∎

Hyperbolic quadratics have many interesting properties [25, section 31].

THEOREM 2.3. *Let the $n \times n$ quadratic $Q(\lambda) = \lambda^2 A + \lambda B + C$ be hyperbolic.*

(a) *The $2n$ eigenvalues of $Q(\lambda)$ are all real and semisimple.*

(b) *There is a gap between the n largest and n smallest eigenvalues, that is, the eigenvalues can be ordered $\lambda_1 \geq \cdots \geq \lambda_n > \lambda_{n+1} \geq \cdots \geq \lambda_{2n}$.*

(c) *$Q(\mu) < 0$ for all $\mu \in (\lambda_{n+1}, \lambda_n)$ and $Q(\mu) > 0$ for all $\mu \in (-\infty, \lambda_{2n}) \cup (\lambda_1, \infty)$.*

(d) *There are n linearly independent eigenvectors associated with the n largest eigenvalues and likewise for the n smallest eigenvalues.*

(e) *The quadratic matrix equation $Q(X) = 0$ in (1.4) has a solvent $S^{(1)}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$ and a solvent $S^{(2)}$ with eigenvalues $\lambda_{n+1}, \ldots, \lambda_{2n}$. Moreover,*

$$Q(\lambda) = \big(\lambda I - S^{(2)^*}\big) A (\lambda I - S^{(1)}) = \big(\lambda I - S^{(1)^*}\big) A (\lambda I - S^{(2)}).$$

The n largest eigenvalues of a hyperbolic quadratic are called the primary eigenvalues and the n smallest eigenvalues are the secondary eigenvalues. The solvents

[1] An eigenvalue of a matrix polynomial $P(\lambda) = \sum_{k=0}^{\ell} \lambda^k P_k$ is semisimple if it appears only in 1×1 Jordan blocks in a Jordan triple for P [7].

$S^{(1)}$ and $S^{(2)}$ having as their eigenvalues the primary eigenvalues and the secondary eigenvalues, respectively, are referred to as the *primary* and *secondary solvents*.

Hyperbolicity can also be defined for matrix polynomials P of arbitrary degree [25, section 31]. The notion has recently been extended in [16] by replacing the assumption of a positive definite leading coefficient matrix with $P(\omega) > 0$ for some $\omega \in \mathbb{R} \cup \{\infty\}$.

The next result gives some characterizations of an overdamped quadratic. First, we need a simple lemma.

LEMMA 2.4. *Let $Q(\lambda) = \lambda^2 A + \lambda B + C$ be Hermitian and let $\mu > 0$. Then $Q(-\mu) < 0$ if and only if $B > \mu A + \mu^{-1} C$.*

Proof. The proof is immediate from $Q(-\mu) = \mu^2 A - \mu B + C < 0 \Leftrightarrow \mu A - B + \mu^{-1} C < 0$. ∎

THEOREM 2.5. *Let $Q(\lambda) = \lambda^2 A + \lambda B + C$ be Hermitian with $A > 0$. Then the following statements are equivalent:*

(a) *$Q(\lambda)$ is overdamped.*

(b) *$Q(\lambda)$ is hyperbolic and all of its eigenvalues are real and nonpositive.*

(c) *$B > 0$, $C \geq 0$, and $B > \mu A + \mu^{-1} C$ for some $\mu > 0$.*

Proof. (a) $\Leftrightarrow$ (b) is proved in [9, Theorem 5]. (b) $\Rightarrow$ (c): By Theorem 2.3(c), $Q(\widetilde{\mu}) < 0$ for some $\widetilde{\mu} < 0$; (c) follows on invoking Lemma 2.4. (c) $\Rightarrow$ (a): $B > \mu A + \mu^{-1} C$ with $\mu > 0$ implies $Q(-\mu) < 0$ by Lemma 2.4, which implies Q is hyperbolic by Theorem 2.2(d) and hence overdamped since $B > 0$ and $C \geq 0$. ∎

It follows from (b) in Theorem 2.5 that if we know an upper bound, say, θ, on the largest eigenvalue λ_1 of a hyperbolic quadratic Q then, with $\lambda = \mu + \theta$, the quadratic Q_θ defined by

$$\begin{aligned} Q(\lambda) = Q(\mu + \theta) &= \mu^2 A + \mu(B + 2\theta A) + C + \theta B + \theta^2 A \\ &= \mu^2 A_\theta + \mu B_\theta + C_\theta \\ &=: Q_\theta(\mu) \end{aligned} \tag{2.2}$$

is overdamped. Thus any hyperbolic quadratic can be transformed into an overdamped quadratic by an appropriate shifting of the eigenvalues. Hence for the purposes of testing hyperbolicity and overdamping it suffices to consider overdamping. We make this restriction in the next two sections and consider in section 5 how to implement the shifting in practice.

3. An iteration for testing overdamping. Suppose we have a Hermitian quadratic $Q(\lambda) = \lambda^2 A + \lambda B + C$, where we assume throughout this section that $A > 0$, $B > 0$, and $C \geq 0$. The challenge is how to test the hyperbolicity (or, equivalently, the overdamping) condition (1.2) or, equivalently, condition (c) in Theorem 2.5.

The primary and secondary solvents $S^{(1)}$ and $S^{(2)}$ of an overdamped quadratic can be found efficiently by applying an iteration based on cyclic reduction [2], [9]. The iteration is

$$\begin{aligned} &S_0 = B, \quad A_0 = A, \quad B_0 = B, \quad C_0 = C, \\ &S_{k+1} = S_k - A_k B_k^{-1} C_k, \\ &A_{k+1} = A_k B_k^{-1} A_k, \\ &B_{k+1} = B_k - A_k B_k^{-1} C_k - C_k B_k^{-1} A_k, \\ &C_{k+1} = C_k B_k^{-1} C_k. \end{aligned} \tag{3.1}$$

The next theorem summarizes properties of the iteration proved in [9, Lemma 6, Theorem 7 and proof].

THEOREM 3.1. *Let $Q(\lambda) = \lambda^2 A + \lambda B + C$ be an $n \times n$ overdamped quadratic with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n > \lambda_{n+1} \geq \cdots \geq \lambda_{2n}$. Consider iteration* (3.1) *and any matrix norm $\|\cdot\|$.*

(a) *The iterates satisfy $A_k > 0$, $C_k \geq 0$, and $B_k > 0$ for all $k \geq 0$.*

(b) *$\|A_k\|\,\|C_k\|$ converges quadratically to zero with*

$$\limsup_{k\to\infty} \sqrt[2^k]{\|A_k\|\,\|C_k\|} \leq \frac{\lambda_n}{\lambda_{n+1}} < 1.$$

(c) *S_k converges quadratically to a nonsingular matrix $\widehat{S}$ with*

$$\limsup_{k\to\infty} \sqrt[2^k]{\|S_k - \widehat{S}\|} \leq \frac{\lambda_n}{\lambda_{n+1}} < 1. \tag{3.2}$$

(d) *The primary and secondary solvents of $Q(X)$, $S^{(1)}$ and $S^{(2)}$, respectively, are given by*

$$S^{(1)} = -\widehat{S}^{-1}C, \quad S^{(2)} = -A^{-1}\widehat{S}^*. \tag{3.3}$$

The next lemma reveals a crucial property of iteration (3.1) for overdamped quadratics. The "only if" part of the result is [9, Lemma 6].

LEMMA 3.2. *Let $\mu > 0$ and assume $A_k > 0$ and $C_k \geq 0$. In* (3.1)*, $B_k > \mu^{2^k} A_k + \mu^{-2^k} C_k$ if and only if $A_{k+1} > 0$, $C_{k+1} \geq 0$, and $B_{k+1} > \mu^{2^{k+1}} A_{k+1} + \mu^{-2^{k+1}} C_{k+1}$.*

Proof. "$\Rightarrow$": We have

$$\begin{aligned} B_{k+1} &= B_k - A_k B_k^{-1} C_k - C_k B_k^{-1} A_k \\ &= B_k - \big(\mu^{2^k} A_k + \mu^{-2^k} C_k\big) B_k^{-1} \big(\mu^{2^k} A_k + \mu^{-2^k} C_k\big) \\ &\qquad + \mu^{2^{k+1}} A_k B_k^{-1} A_k + \mu^{-2^{k+1}} C_k B_k^{-1} C_k \\ &> \mu^{2^{k+1}} A_k B_k^{-1} A_k + \mu^{-2^{k+1}} C_k B_k^{-1} C_k, \end{aligned}$$

where we have used the fact that $X - YX^{-1}Y > Y - YY^{-1}Y = 0$ when $X > Y > 0$. Clearly, $A_{k+1} > 0$ and $C_{k+1} \geq 0$ since $B_k^{-1} > 0$.

"$\Leftarrow$": As in the first part we have

$$B_{k+1} = B_k - F_k B_k^{-1} F_k + F_{k+1}, \tag{3.4}$$

where $F_k = \mu^{2^k} A_k + \mu^{-2^k} C_k$. Now if $B_{k+1} > \mu^{2^{k+1}} A_{k+1} + \mu^{-2^{k+1}} C_{k+1} = F_{k+1}$ then (3.4) gives $B_k - F_k B_k^{-1} F_k > 0$. Note that $B_k - F_k B_k^{-1} F_k$ is the Schur complement of $B_k > 0$ in

$$T = \begin{bmatrix} B_k & F_k \\ F_k & B_k \end{bmatrix}.$$

So we have $T > 0$, and it follows that $B_k - F_k > 0$ (for example, by looking at the $(1,1)$ block of the congruence $\left[\begin{smallmatrix} I & -I \\ 0 & I \end{smallmatrix}\right] T \left[\begin{smallmatrix} I & 0 \\ -I & I \end{smallmatrix}\right]$). Therefore $B_k > F_k = \mu^{2^k} A_k + \mu^{-2^k} C_k$. ∎

In view of Theorem 2.5(c), Lemma 3.2 implies that Q is overdamped *if and only if* any one of the quadratics

$$Q_k(\lambda) = \lambda^2 A_k + \lambda B_k + C_k \tag{3.5}$$

generated during the iteration is overdamped, assuming that $A_k > 0$ and $C_k \geq 0$ for all k. Note that the latter assumption holds if $B_k > 0$ for all k.

COROLLARY 3.3. *Let Q be a Hermitian quadratic with $A, B > 0$ and $C \geq 0$. For iteration* (3.1) *and any fixed $m \geq 0$, if $B_k > 0$ for $k = 1\colon m-1$ and*

$$B_m > \mu^{2^m} A_m + \mu^{-2^m} C_m \tag{3.6}$$

for some $\mu > 0$, then $B > \mu A + \mu^{-1} C$ and Q is overdamped.

Intuitively, we can think of the scalars μ^{2^m} and μ^{-2^m} in (3.6) as trying to balance A_m and C_m. This suggests that (3.6) could be replaced by $B_m > \widetilde{A}_m + \widetilde{C}_m$ if the iteration is scaled so that $\widetilde{A}_m$ and $\widetilde{C}_m$ are balanced. Normwise balancing is included in the following scaled version of (3.1), introduced in [9]; it generates iterates $\widetilde{A}_k$, B_k (unchanged from (3.1)), and $\widetilde{C}_k$ according to

$$\begin{aligned}
&\alpha_0 = \sqrt{\|C\|/\|A\|},\\
&\widetilde{A}_0 = \alpha_0 A, \quad B_0 = B, \quad \widetilde{C}_0 = \alpha_0^{-1} C,\\
&A_{k+1} = \widetilde{A}_k B_k^{-1} \widetilde{A}_k,\\
&B_{k+1} = B_k - \widetilde{A}_k B_k^{-1} \widetilde{C}_k - \widetilde{C}_k B_k^{-1} \widetilde{A}_k,\\
&C_{k+1} = \widetilde{C}_k B_k^{-1} \widetilde{C}_k,\\
&\alpha_{k+1} = \sqrt{\|C_{k+1}\|/\|A_{k+1}\|},\\
&\widetilde{A}_{k+1} = \alpha_{k+1} A_{k+1}, \quad \widetilde{C}_{k+1} = \alpha_{k+1}^{-1} C_{k+1}.
\end{aligned} \tag{3.7}$$

Here we have assumed that $C \neq 0$ (the overdamping condition holds for the trivial case $C = 0$ by (1.2)); thus $\alpha_k > 0$ for each $k \geq 0$. The scaling procedure ensures that $\|\widetilde{A}_k\| = \|\widetilde{C}_k\|$ and $\|\widetilde{A}_k\|\,\|\widetilde{C}_k\| = \|A_k\|\,\|C_k\|$.

The next result describes the behavior of the scaled iteration.

THEOREM 3.4. *A Hermitian quadratic Q with $A, B > 0$ and $0 \neq C \geq 0$ is overdamped if and only if in* (3.7)

$$B_k > 0 \text{ for all } k, \quad \lim_{k\to\infty} \widetilde{A}_k = 0, \quad \lim_{k\to\infty} \widetilde{C}_k = 0, \quad \lim_{k\to\infty} B_k > 0, \tag{3.8}$$

and in this case

$$\limsup_{k\to\infty} \sqrt[2^k]{\|\widetilde{A}_k\|} = \limsup_{k\to\infty} \sqrt[2^k]{\|\widetilde{C}_k\|} \leq \left(\frac{\lambda_n}{\lambda_{n+1}}\right)^{1/2}, \tag{3.9}$$

$$\limsup_{k\to\infty} \sqrt[2^k]{\|B_k - \widehat{B}\|} \leq \frac{\lambda_n}{\lambda_{n+1}}, \tag{3.10}$$

with $\widehat{B} = A(S^{(1)} - S^{(2)})$.

Proof. Assume that the conditions in (3.8) hold. Then $B_m > \widetilde{A}_m + \widetilde{C}_m$ for some $m \geq 0$. It is easy to see that the iterates $\widetilde{A}_k$ and $\widetilde{C}_k$ defined in (3.7) are related to A_k and C_k in (3.1) by

$$\widetilde{A}_k = \alpha_0^{2^k} \alpha_1^{2^{k-1}} \dots \alpha_{k-1}^2 \alpha_k A_k, \quad \widetilde{C}_k = \alpha_0^{-2^k} \alpha_1^{-2^{k-1}} \dots \alpha_{k-1}^{-2} \alpha_k^{-1} C_k, \quad k \geq 0.$$

So $B_m > \widetilde{A}_m + \widetilde{C}_m$ implies $B_m > \mu^{2^m} A_m + \mu^{-2^m} C_m$ with $\mu = \alpha_0 \alpha_1^{2^{-1}} \alpha_2^{2^{-2}} \dots \alpha_m^{2^{-m}}$, which implies Q is overdamped by Corollary 3.3.

TABLE 3.1
Number of iterations m to verify that the quadratic defined by (3.11) *is overdamped.*

β	1	0.62	0.61	0.53	0.52	0.5197	0.519616	0.51961525	0.5196152423
m	0	0	1	1	2	3	5	8	12

TABLE 3.2
Number of iterations m to verify that the quadratic defined by (3.11) *is* not *overdamped.*

β	0.36	0.47	0.50	0.51	0.5196	0.519615	0.51961524	0.5196152422
m	1	2	3	4	8	11	15	17

Now assume the QEP is overdamped. Then, from Theorem 3.1(a), $B_k > 0$ for each $k \geq 0$, while, since $\|\widetilde{A}_k\| = \|\widetilde{C}_k\| = (\|A_k\|\|C_k\|)^{1/2}$, Theorem 3.1(b) implies $\lim \widetilde{A}_k = \lim \widetilde{C}_k = 0$ and that (3.9) holds. To show the convergence of B_k, we note that from (3.1), $B_{k+1} = B_k - (S_k - S_{k+1}) - (S_k - S_{k+1})^*$, which implies

$$B_k = B_0 - (S_0 - S_k) - (S_0 - S_k)^* = -B + S_k + S_k^*.$$

In view of (3.2), (3.3), and $B_k > 0$, (3.10) holds with $\widehat{B} = -B + \widehat{S} + \widehat{S}^* = A(S^{(1)} - S^{(2)}) \geq 0$. Since the sequence $\{\|B_k^{-1}\|\}$ is known to be bounded (see the proof of [9, Theorem 7]), we have $\widehat{B} > 0$. ∎

The next result confirms that μ can be removed from (3.6) for the scaled iteration. It follows readily from Theorem 3.4 and its proof.

COROLLARY 3.5. *A Hermitian quadratic Q with $A, B > 0$ and $0 \neq C \geq 0$ is overdamped if and only if, for some $m \geq 0$, $B_k > 0$ for $k = 1\colon m-1$ in* (3.7) *and $B_m > \widetilde{A}_m + \widetilde{C}_m$.*

The corollary is important for two reasons. First, it provides a basis for an elegant, practical test for overdamping, as definiteness of a matrix is easily tested numerically. Second, in the case of an affirmative test result a μ with $Q(-\mu) < 0$ can be identified, and such a μ is very useful when we go on to solve the QEP, as we will show in section 5.

From a numerical point of view it is preferable to work with the original data as much as possible. The following variant of Corollary 3.5 tests the overdamping condition using the original quadratic Q and will be the basis of the algorithm in section 5. It follows readily from Corollary 3.3 and Theorem 3.4 and its proof.

COROLLARY 3.6. *A Hermitian quadratic Q with $A, B > 0$ and $0 \neq C \geq 0$ is overdamped if and only if, for some $m \geq 0$, $B_k > 0$ for $k = 1\colon m-1$ in* (3.7) *and $Q(-\mu_m) < 0$, where $\mu_m = \alpha_0 \alpha_1^{2^{-1}} \alpha_2^{2^{-2}} \cdots \alpha_m^{2^{-m}} > 0$ and the α_k are defined in* (3.7).

Usually, only a few iterations of the cyclic reduction algorithm (3.7) will be necessary. To illustrate, we consider a quadratic $Q(\lambda)$ of dimension $n = 100$ defined by

$$(3.11)\quad A = I,\ B = \beta \begin{bmatrix} 20 & -10 & & & \\ -10 & 30 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 30 & -10 \\ & & & -10 & 20 \end{bmatrix},\ C = \begin{bmatrix} 15 & -5 & & \\ -5 & 15 & \ddots & \\ & \ddots & \ddots & -5 \\ & & -5 & 15 \end{bmatrix},$$

where $\beta > 0$ is a real parameter. This example, which comes from a damped mass-spring system, is used in [13] with $\beta = 1$. We use the 1-norm in (3.7). Tables 3.1 and 3.2 report the number of iterations required to demonstrate that Q is over-

damped, through verification of the conditions in Corollary 3.6, or that it is not overdamped, through generation of a non-positive definite iterate B_m. Note that when Q is "strongly" overdamped and when Q is far from being overdamped, the overdamping condition is shown to hold or not after just a few iterations.

4. Convergence analysis for weakly overdamped quadratics. For the example at the end of section 3 and some $\beta_0 \in (0.5196152422, 0.5196152423)$, the inequality (1.2) holds as a weak inequality with equality attained for some nonzero x. We have seen that the overdamping test requires a very small number of iterations when β is not close to β_0. When $\beta \approx \beta_0$, the number of iterations increases but is still under 20 in our experiments. The purpose of this section is to explain this behavior by showing that the convergence of iteration (3.1) is reasonably fast even when the QEP is weakly overdamped in the sense defined as follows.

DEFINITION 4.1. *$Q(\lambda)$ is weakly hyperbolic if A, B, and C are Hermitian, $A > 0$, and*

$$\gamma = \min_{\|x\|_2=1} [(x^*Bx)^2 - 4(x^*Ax)(x^*Cx)] \geq 0. \tag{4.1}$$

DEFINITION 4.2. *$Q(\lambda)$ is weakly overdamped if it is weakly hyperbolic with $B > 0$ and $C \geq 0$.*

The eigenvalues of a weakly hyperbolic Q are real and those of a weakly overdamped Q are real and nonpositive. The following result collects further properties of a weakly overdamped quadratic [25, section 31].

THEOREM 4.3. *Let $Q(\lambda) = \lambda^2 A + \lambda B + C$ be a weakly overdamped $n \times n$ quadratic.*

(a) *If $\gamma = 0$ in (4.1), then $Q(\lambda)$ has $2n$ real eigenvalues that can be ordered $\lambda_1 \geq \cdots \geq \lambda_n = \lambda_{n+1} \geq \cdots \geq \lambda_{2n}$. The partial multiplicities[2] of λ_n are at most 2, and the eigenvalues other than λ_n are semisimple.*

(b) *$Q(\lambda_n) \leq 0$.*

(c) *The quadratic matrix equation $Q(X) = 0$ in (1.4) has a solvent $S^{(1)}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$ and a solvent $S^{(2)}$ with eigenvalues $\lambda_{n+1}, \ldots, \lambda_{2n}$.*

In the overdamped case considered in the previous section, convergence results for the iteration (3.1) are established using matrix identities obtained from the cyclic reduction method. Those identities do not contain enough information about (3.1) to allow a proof of convergence for weakly overdamped quadratics with $\gamma = 0$, for which $\lambda_{n+1} = \lambda_n$. We now study this critical case and thereby obtain a better understanding of the convergence of the iteration for overdamped QEPs with $\lambda_n \approx \lambda_{n+1}$. The next lemma shows that (3.1) remains well defined in the critical case, which is the setting for the rest of this section.

LEMMA 4.4. *For a weakly overdamped quadratic $Q(\lambda) = \lambda^2 A + \lambda B + C$ with $\gamma = 0$ in (4.1), there is a positive real number μ such that for the iteration (3.1)*

$$A_k > 0, \quad C_k \geq 0, \quad B_k \geq \mu^{2^k} A_k + \mu^{-2^k} C_k \tag{4.2}$$

for all $k \geq 0$.

Proof. We have $\lambda_n \leq \lambda_1 \leq 0$. If $\lambda_n = 0$ then, from Theorem 4.3, $C = Q(\lambda_n) \leq 0$. Since $C \geq 0$ we must have $C = 0$. However, $\gamma > 0$ for the trivial case $C = 0$. Therefore $\lambda_n < 0$ since $\gamma = 0$. It then follows from $Q(\lambda_n) \leq 0$ that $B \geq \mu A + \mu^{-1} C$

[2]The partial multiplicities of an eigenvalue of Q are the sizes of the Jordan blocks in which it appears in a Jordan triple for Q [7].

for $\mu = -\lambda_n > 0$. The inequalities in (4.2) are then proved inductively using the technique from the proof of the first part of Lemma 3.2. ∎

Lin and Xu [24] recently showed that Meini's iterations based on cyclic reduction for the matrix equation $X + A^*X^{-1}A = Q$ [26] can also be derived from a structure-preserving doubling algorithm. Following their approach we show that the iteration (3.1) is related to a doubling algorithm, and we use this observation to study the convergence of (3.1) for weakly overdamped quadratics. The rate of convergence will be shown to be at least linear with constant 1/2 in the generic case, which is the case where $\lambda_n = \lambda_{n+1}$ is a multiple eigenvalue with partial multiplicities all equal to 2 (that is, λ_n occurs only in 2×2 Jordan blocks). This rate and constant are to be expected in view of the results of Guo in [8].

LEMMA 4.5. *Let* $X = \left[\begin{smallmatrix} A & 0 \\ H & -I \end{smallmatrix}\right]$ *and* $Y = \left[\begin{smallmatrix} G & I \\ C & 0 \end{smallmatrix}\right]$ *be block* 2×2 *matrices with* $n \times n$ *blocks. When* $H + G$ *is nonsingular there exist* $2n \times 2n$ *matrices* $\widetilde{X}$ *and* $\widetilde{Y}$ *such that* (a) $\widetilde{X}Y = \widetilde{Y}X$ *and* (b) $\widetilde{X}X$, $\widetilde{Y}Y$ *have the same zero and identity blocks as* X *and* Y, *respectively.*

Proof. Applying block row permutations and block Gaussian elimination to $\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right]$ yields $P\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right] = \left[\begin{smallmatrix} U \\ 0 \end{smallmatrix}\right]$, where $U = \left[\begin{smallmatrix} G & I \\ G+H & 0 \end{smallmatrix}\right]$ and

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} 0 & 0 & I & 0 \\ 0 & I & I & 0 \\ \hline I & -A(G+H)^{-1} & -A(G+H)^{-1} & 0 \\ 0 & C(G+H)^{-1} & C(G+H)^{-1} & -I \end{array}\right].$$

Since $\left[\begin{smallmatrix} P_{21} & P_{22} \end{smallmatrix}\right]\left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right] = 0$, the required equality $\widetilde{Y}X = \widetilde{X}Y$ is satisfied with $\widetilde{X} := -P_{22}$ and $\widetilde{Y} := P_{21}$. Furthermore,

$$\widetilde{X}X = \begin{bmatrix} A(G+H)^{-1}A & 0 \\ H - C(G+H)^{-1}A & -I \end{bmatrix}, \qquad \widetilde{Y}Y = \begin{bmatrix} G - A(G+H)^{-1}C & I \\ C(G+H)^{-1}C & 0 \end{bmatrix}. \qquad \blacksquare$$

Lemma 4.5 and its proof suggest the recurrence

$$X_{k+1} = \widetilde{X}_k X_k, \quad Y_{k+1} = \widetilde{Y}_k Y_k, \qquad k \geq 0, \tag{4.3}$$

with

$$X_k = \begin{bmatrix} A_k & 0 \\ H_k & -I \end{bmatrix}, \quad Y_k = \begin{bmatrix} G_k & I \\ C_k & 0 \end{bmatrix} \tag{4.4}$$

and

$$\widetilde{X}_k = \begin{bmatrix} A_k(G_k+H_k)^{-1} & 0 \\ -C_k(G_k+H_k)^{-1} & I \end{bmatrix}, \quad \widetilde{Y}_k = \begin{bmatrix} I & -A_k(G_k+H_k)^{-1} \\ 0 & C_k(G_k+H_k)^{-1} \end{bmatrix},$$

which leads to

$$\begin{aligned} A_{k+1} &= A_k(G_k+H_k)^{-1}A_k, \\ G_{k+1} &= G_k - A_k(G_k+H_k)^{-1}C_k, \\ H_{k+1} &= H_k - C_k(G_k+H_k)^{-1}A_k, \\ C_{k+1} &= C_k(G_k+H_k)^{-1}C_k. \end{aligned} \tag{4.5}$$

With

$$A_0 = A, \quad C_0 = C, \quad G_0 = 0, \quad H_0 = B, \tag{4.6}$$

the iteration (3.1) is recovered from (4.5) by letting $B_k = G_k + H_k$ and $S_k = H_k^*$. By Lemma 4.4, $B_k > 0$ for all $k \geq 0$. Therefore with the starting matrices (4.6), iteration (4.5) is well defined. Note that X_k in (4.4) is nonsingular for all $k \geq 0$ and, from (4.3) and the property that $\widetilde{X}_k Y_k = \widetilde{Y}_k X_k$,

$$X_{k+1}^{-1} Y_{k+1} = (\widetilde{X}_k X_k)^{-1} \widetilde{Y}_k Y_k = X_k^{-1} \widetilde{X}_k^{-1} \widetilde{Y}_k Y_k = X_k^{-1} Y_k X_k^{-1} Y_k = (X_k^{-1} Y_k)^2. \tag{4.7}$$

It follows from (4.7) that for all $k \geq 0$,

$$X_k^{-1} Y_k = \left(X_0^{-1} Y_0\right)^{2^k}. \tag{4.8}$$

The identity (4.8) is what we need to prove the convergence of (4.5) with (4.6) and hence the convergence of (3.1).

The next result describes the convergence behavior in the generic case.

THEOREM 4.6. *Let $Q(\lambda)$ be weakly overdamped with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n = \lambda_{n+1} \geq \cdots \geq \lambda_{2n}$, and assume that the partial multiplicities of λ_n are all equal to 2. Let $S^{(1)}$ and $S^{(2)}$ be the primary and secondary solvents of $Q(X) = 0$, respectively, and assume that λ_n is a semisimple eigenvalue of $S^{(1)}$ and $S^{(2)}$. Then the iterates G_k, H_k, A_k, and C_k defined by* (4.5) *and* (4.6) *satisfy*

$$\limsup_{k\to\infty} \sqrt[k]{\|G_k - AS^{(1)}\|} \leq \frac{1}{2}, \quad \limsup_{k\to\infty} \sqrt[k]{\|H_k + AS^{(2)}\|} \leq \frac{1}{2},$$
$$\limsup_{k\to\infty} \sqrt[k]{\|A_k\|\|C_k\|} \leq \frac{1}{4}.$$

Proof. We start by making the change of variables (or scaling) $\lambda = \mu\theta$, where $\theta = |\lambda_n| > 0$ (see the proof of Lemma 4.4) so that $\mu_n = \mu_{n+1} = -1$, and we define $\widehat{Q}(\mu) = \mu^2 \widehat{A} + \mu \widehat{B} + \widehat{C}$ with $(\widehat{A}, \widehat{B}, \widehat{C}) = (\theta A, B, \theta^{-1} C)$. For this triple denote the iterates of (4.5) by $\widehat{A}_k$, $\widehat{G}_k$, $\widehat{H}_k$, and $\widehat{C}_k$. It is easy to see that for all $k \geq 0$, $\widehat{G}_k = G_k$, $\widehat{H}_k = H_k$, $\widehat{A}_k = \theta^{2^k} A_k$, and $\widehat{C}_k = \theta^{-2^k} C_k$ so that $\|A_k\|\,\|C_k\| = \|\widehat{A}_k\|\,\|\widehat{C}_k\|$. The primary and secondary solvents of $\widehat{A}S^2 + \widehat{B}S + \widehat{C} = 0$ are $\widehat{S}^{(1)} = \theta^{-1} S^{(1)}$ and $\widehat{S}^{(2)} = \theta^{-1} S^{(2)}$, respectively. Note that $\widehat{A}\widehat{S}^{(i)} = AS^{(i)}$, $i = 1, 2$. To avoid notational clutter, we omit the hats on matrices in the rest of the proof.

We now consider the iterations for the block 2×2 matrices X_k and Y_k in (4.4). With $A_0 = A$, $C_0 = C$, $G_0 = 0$, and $H_0 = B$, the pencil

$$\mu X_0 + Y_0 = \mu \begin{bmatrix} A & 0 \\ B & -I_n \end{bmatrix} + \begin{bmatrix} 0 & I_n \\ C & 0 \end{bmatrix} \tag{4.9}$$

is a linearization of $Q(\mu)$ [7]. Hence $-X_0^{-1} Y_0$ and $Q(\mu)$ have the same eigenvalues, with the same partial multiplicities. Suppose there are r 2×2 Jordan blocks associated with eigenvalues equal to $\mu_n = -1$, where $r \geq 1$ by assumption. Rearranging the Jordan canonical form of $X_0^{-1} Y_0$ appropriately yields

$$V^{-1}(X_0^{-1} Y_0) V = \begin{bmatrix} D_2 \oplus I_r & 0 \oplus I_r \\ 0 & D_1 \oplus I_r \end{bmatrix} =: D_V, \tag{4.10}$$

$$W^{-1}(X_0^{-1} Y_0) W = \begin{bmatrix} D_2 \oplus I_r & 0 \\ 0 \oplus I_r & D_1 \oplus I_r \end{bmatrix} =: D_W, \tag{4.11}$$

where V and W are nonsingular, D_1 and D_2 are $(n-r) \times (n-r)$ diagonal matrices containing the (semisimple) eigenvalues less than 1 and greater than 1 in modulus,

respectively, and $M \oplus N$ denotes $\left[\begin{smallmatrix} M & 0 \\ 0 & N \end{smallmatrix}\right]$. Now partition V and W as block 2×2 matrices with $n \times n$ blocks

$$V = \begin{bmatrix} V_1 & V_3 \\ V_2 & V_4 \end{bmatrix}, \qquad W = \begin{bmatrix} W_1 & W_3 \\ W_2 & W_4 \end{bmatrix},$$

and note from (4.10)–(4.11) that

$$X_0^{-1}Y_0 \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} (D_2 \oplus I_r), \qquad X_0^{-1}Y_0 \begin{bmatrix} W_3 \\ W_4 \end{bmatrix} = \begin{bmatrix} W_3 \\ W_4 \end{bmatrix} (D_1 \oplus I_r). \tag{4.12}$$

By Theorem 4.3 and our assumption on $S^{(1)}$ and $S^{(2)}$ there exist nonsingular U_1 and U_2 such that

$$-S^{(1)} = U_1(D_1 \oplus I_r)U_1^{-1}, \qquad -S^{(2)} = U_2(D_2 \oplus I_r)U_2^{-1}. \tag{4.13}$$

Since $S^{(i)}$, $i = 1, 2$, is a solution of $Q(X) = 0$, from (4.9) we obtain

$$X_0^{-1}Y_0 \begin{bmatrix} I_n \\ -AS^{(i)} \end{bmatrix} = \begin{bmatrix} I_n \\ -AS^{(i)} \end{bmatrix} (-S^{(i)}), \quad i = 1, 2.$$

On comparing with the invariant subspaces in (4.12) and using (4.13) we deduce that

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} U_2 \\ -AS^{(2)}U_2 \end{bmatrix} Z_1, \qquad \begin{bmatrix} W_3 \\ W_4 \end{bmatrix} = \begin{bmatrix} U_1 \\ -AS^{(1)}U_1 \end{bmatrix} Z_2,$$

with Z_1 and Z_2 nonsingular, where we have also used the fact that there are exactly r eigenvectors of $X_0^{-1}Y_0$ corresponding to the eigenvalue 1. Hence V_1 and W_3 are nonsingular and

$$-AS^{(2)} = V_2V_1^{-1}, \quad -AS^{(1)} = W_4W_3^{-1}. \tag{4.14}$$

By (4.8)–(4.11) we have $V^{-1}(X_k^{-1}Y_k)V = D_V^{2^k}$ and $W^{-1}(X_k^{-1}Y_k)W = D_W^{2^k}$, so that

$$Y_kV = X_kVD_V^{2^k}, \quad Y_kW = X_kWD_W^{2^k}. \tag{4.15}$$

On equating blocks using (4.4) this yields

$$G_kV_1 + V_2 = A_kV_1(D_2^{2^k} \oplus I_r), \tag{4.16}$$

$$G_kV_3 + V_4 = A_kV_1(0 \oplus 2^kI_r) + A_kV_3(D_1^{2^k} \oplus I_r), \tag{4.17}$$

$$C_kV_1 = (H_kV_1 - V_2)(D_2^{2^k} \oplus I_r), \tag{4.18}$$

$$C_kV_3 = (H_kV_1 - V_2)(0 \oplus 2^kI_r) + (H_kV_3 - V_4)(D_1^{2^k} \oplus I_r) \tag{4.19}$$

and

$$G_kW_1 + W_2 = A_kW_1(D_2^{2^k} \oplus I_r) + A_kW_3(0 \oplus 2^kI_r), \tag{4.20}$$

$$G_kW_3 + W_4 = A_kW_3(D_1^{2^k} \oplus I_r), \tag{4.21}$$

$$C_kW_1 = (H_kW_1 - W_2)(D_2^{2^k} \oplus I_r) + (H_kW_3 - W_4)(0 \oplus 2^kI_r), \tag{4.22}$$

$$C_kW_3 = (H_kW_3 - W_4)(D_1^{2^k} \oplus I_r). \tag{4.23}$$

By (4.22) and (4.23) we have

$$C_k(W_3 - W_1(0 \oplus 2^{-k} I_r)) = (H_k W_3 - W_4)(D_1^{2^k} \oplus 0) - (H_k W_1 - W_2)(0 \oplus 2^{-k} I_r). \tag{4.24}$$

By (4.18) we have

$$H_k = V_2 V_1^{-1} + C_k V_1 (D_2^{-2^k} \oplus I_r) V_1^{-1}. \tag{4.25}$$

Inserting (4.25) in (4.24) we obtain

$$\begin{aligned} C_k \Big(W_3 - W_1(0 \oplus 2^{-k} I_r) - V_1 (D_2^{-2^k} \oplus I_r) V_1^{-1} \big(W_3 (D_1^{2^k} \oplus 0) - W_1 (0 \oplus 2^{-k} I_r) \big) \Big) \\ = (V_2 V_1^{-1} W_3 - W_4)(D_1^{2^k} \oplus 0) - (V_2 V_1^{-1} W_1 - W_2)(0 \oplus 2^{-k} I_r), \end{aligned}$$

from which it follows, since D_1 and D_2 are diagonal with diagonal elements of magnitude less than 1 and greater than 1, respectively, that

$$C_k = O(2^{-k}); \tag{4.26}$$

the latter notation means that $\|C_k\| = O(2^{-k})$. It then follows from (4.25) and (4.14) that

$$H_k + AS^{(2)} = H_k - V_2 V_1^{-1} = O(2^{-k}). \tag{4.27}$$

By (4.20) and (4.21),

$$G_k W_3 + W_4 - (G_k W_1 + W_2)(0 \oplus 2^{-k} I_r) = A_k (W_3 (D_1^{2^k} \oplus 0) - W_1 (0 \oplus 2^{-k} I_r)). \tag{4.28}$$

By (4.16),

$$A_k = (G_k V_1 + V_2)(D_2^{-2^k} \oplus I_r) V_1^{-1}. \tag{4.29}$$

Inserting (4.29) in (4.28) we obtain

$$G_k W_3 + W_4 - (G_k W_1 + W_2)(0 \oplus 2^{-k} I_r) = (G_k V_1 + V_2) M_k,$$

with $M_k = O(2^{-k})$. Thus

$$-G_k \big(W_3 - W_1 (0 \oplus 2^{-k} I_r) - V_1 M_k \big) = W_4 - W_2 (0 \oplus 2^{-k} I_r) - V_2 M_k.$$

It follows from (4.14) that

$$G_k - AS^{(1)} = G_k + W_4 W_3^{-1} = O(2^{-k}). \tag{4.30}$$

Postmultiplying (4.16) by $D_2^{-2^k} \oplus 0$ gives

$$(G_k V_1 + V_2)(D_2^{-2^k} \oplus 0) = A_k V_1 (I_r \oplus 0), \tag{4.31}$$

while postmultiplying (4.17) by $0 \oplus 2^{-k} I_r$ gives

$$(G_k V_3 + V_4)(0 \oplus 2^{-k} I_r) = A_k V_1 (0 \oplus I_r) + A_k V_3 (0 \oplus 2^{-k} I_r). \tag{4.32}$$

Adding (4.31) and (4.32) we get

$$A_k (V_1 + V_3 (0 \oplus 2^{-k} I_r)) = (G_k V_1 + V_2)(D_2^{-2^k} \oplus 0) + (G_k V_3 + V_4)(0 \oplus 2^{-k} I_r).$$

It follows that

$$A_k = O(2^{-k}), \tag{4.33}$$

since $\{G_k\}$ has been shown to be bounded. Equations (4.26), (4.27), (4.30), and (4.33) yield the required convergence results. ∎

For S_k and B_k in iteration (3.1) we obtain the following convergence result.

COROLLARY 4.7. *Under the conditions of Theorem* 4.6*, the iterates* S_k *and* B_k *in* (3.1) *satisfy*

$$\limsup_{k\to\infty} \sqrt[k]{\|S_k - \widehat{S}\|} \le \frac{1}{2}, \qquad \limsup_{k\to\infty} \sqrt[k]{\|B_k - \widehat{B}\|} \le \frac{1}{2},$$

where $\widehat{S} = -S^{(2)^*}A$ *is nonsingular and* $\widehat{B} = A(S^{(1)} - S^{(2)}) \ge 0$ *is singular.*

Proof. The convergence results follow from Theorem 4.6 by noting $B_k = H_k + G_k$ and $S_k = H_k^*$. By (4.27) and (4.30), $\widehat{B} = A(S^{(1)} - S^{(2)})$. We have $\widehat{B} \ge 0$ since $B_k > 0$ for each k, by Lemma 4.4. We now show that $\widehat{B}$ is singular. Using (4.9) it is easy to check that

$$(-X_0^{-1}Y_0)\begin{bmatrix} I & I \\ -AS^{(1)} & -AS^{(2)} \end{bmatrix} = \begin{bmatrix} I & I \\ -AS^{(1)} & -AS^{(2)} \end{bmatrix}(S^{(1)} \oplus S^{(2)}), \tag{4.34}$$

and $S^{(1)} \oplus S^{(2)}$ is diagonalizable. Now $-X_0^{-1}Y_0$ is not diagonalizable, by assumption, since it has at least one eigenvalue of partial multiplicity 2. Thus (4.34) can only hold if $\left[\begin{smallmatrix} I & I \\ -AS^{(1)} & -AS^{(2)} \end{smallmatrix}\right]$ is singular. Thus the Schur complement $\widehat{B} = A(S^{(1)} - S^{(2)})$ is singular. ∎

In the generic case for a weakly overdamped Q with $\gamma = 0$, in which all of the partial multiplicities of λ_n are 2, Q is in some sense irreducible or coupled. The next example shows that this condition is necessary for the conclusions in Theorem 4.6 and Corollary 4.7 (and at the same time answers an open question from [9, section 4]). Consider

$$Q(\lambda) = \lambda^2 A + \lambda B + C = \lambda^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \lambda \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}.$$

It is easy to see that $\gamma = 0$, so $Q(\lambda)$ is weakly overdamped with eigenvalues $\{0, -1, -1, -2\}$ with $\lambda_2 = \lambda_3 = -1$ semisimple. In (3.1) and (4.5), (4.6),

$$\lim_{k\to\infty} A_k = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad \lim_{k\to\infty} B_k = I_2, \quad \lim_{k\to\infty} C_k = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$
$$\lim_{k\to\infty} G_k = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \lim_{k\to\infty} H_k = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}.$$

Neither A_k nor C_k converges to zero. We also note that the convergence is quadratic for B_k, G_k, and H_k. Moreover, B_k converges to a nonsingular matrix. This does not come as a surprise, since $Q(\lambda)$ can be decomposed into the direct sum of two scalar quadratics

$$Q_1(\lambda) = \lambda^2 + 3\lambda + 2, \qquad Q_2(\lambda) = \lambda^2 + \lambda.$$

It is readily seen that Q_1 is overdamped with eigenvalues $-1, -2$ and that Q_2 is overdamped with eigenvalues $0, -1$. Thus the convergence of B_k to a positive definite matrix is guaranteed by Theorem 3.4 applied to each component of the direct sum.

5. Algorithm for the detection and numerical solution. Let $Q(\lambda) = \lambda^2 A + \lambda B + C$ be Hermitian with $A > 0$. We develop in this section an efficient algorithm that checks if Q is hyperbolic and, if it is, computes some or all of the eigenvalues and associated eigenvectors, exploiting the symmetry and hyperbolicity and thereby preserving the spectral properties.

Our algorithm consists of three steps:

1. *Preprocessing.* This step forms $Q_\theta(\lambda) \equiv Q(\lambda + \theta) = \lambda^2 A_\theta + \lambda B_\theta + C_\theta$ with θ such that $B_\theta > 0$ and $C_\theta \geq 0$ or concludes that Q is not hyperbolic and terminates the algorithm.
2. *Overdamping test.* This step checks the overdamping condition for Q_θ. If Q_θ is overdamped, a $\mu \in \mathbb{R}$ such that $Q_\theta(\mu) = Q(\mu + \theta) < 0$ is also computed; otherwise the algorithm terminates.
3. *Solution.* The quadratic Q_θ is converted into a definite pencil $\lambda X + Y \in \mathbb{C}^{2n\times 2n}$ with $X > 0$ or $Y > 0$. The eigenvalues and eigenvectors of $Q(\lambda)$ are then obtained from the eigendecomposition of a $2n \times 2n$ Hermitian matrix obtained by transforming $\lambda X + Y$ and exploiting the definiteness of X or Y and the block structure of X and Y.

We now detail each of these three steps and compare the cost and stability of our solution process with that of three alternative ways of solving the QEP: The QZ algorithm applied to a linearization of $Q(\lambda)$, the J-orthogonal Jacobi algorithm [28] also applied to a linearization of $Q(\lambda)$, and the method of computing the eigenpairs of the primary and secondary solvents obtained via the cyclic reduction method [9].

At different stages our algorithm needs to test the (semi)definiteness of a matrix. This is done by attempting a Cholesky factorization, with complete pivoting in the case of semidefinitiness: Completion of the factorization means the matrix is (semi)definite. This is a numerically stable test, as shown in [10].

5.1. Preprocessing step. The preprocessing step aims to eliminate, by simple tests, quadratics that are not hyperbolic and to produce, if possible, a shifted quadratic $Q_\theta(\lambda) = Q(\lambda+\theta)$ (with $\theta = 0$ is possible) for which the necessary condition

$$B > 0, \quad C \geq 0 \tag{5.1}$$

for overdamping is satisfied.

If B is singular then, by (1.2), Q cannot be hyperbolic. Assume now that B is nonsingular but not positive definite or C is not positive semidefinite. Since $A > 0$, for $\theta > 0$ large enough the matrices

$$B_\theta = B + 2\theta A, \quad C_\theta = C + \theta B + \theta^2 A$$

defining the shifted quadratic $Q_\theta(\lambda) = Q(\lambda+\theta)$ with $A_\theta = A$ (see (2.2)) satisfy (5.1). To avoid numerical instability in the formation of B_θ and C_θ (due to the possibly large variation in $\|A\|$, $\|B\|$, and $\|C\|$) we would ideally like to choose θ close to

$$\theta_{\mathrm{opt}} = \inf\{\theta \in \mathbb{R} : \ B + 2\theta A > 0, \ C + \theta B + \theta^2 A \geq 0\}.$$

Rather than solving this optimization problem we choose θ to be an upper bound on the modulus of λ_1, the right-most eigenvalue of Q. With such a shift, all of the eigenvalues of Q_θ lie in the left half plane. When Q is hyperbolic, Q_θ is also hyperbolic with real and nonpositive eigenvalues. Thus $B_\theta > 0$ and $C_\theta \geq 0$ by Theorem 2.5. Therefore if $B_\theta \not> 0$ or $C_\theta \not\geq 0$ we can conclude that Q is not hyperbolic. If $B_\theta > 0$ and $C_\theta \geq 0$ we proceed to step 2.

TABLE 5.1
Operation count for the preprocessing step. Matrices are assumed real and of dimension n.

Operations		Cost (flops)
Cholesky factorization of B and C to check definiteness.		$2n^3/3$ or less
Computation of θ when B and/or C not positive definite:		
Cholesky factorization of A.		$n^3/3$
$\|A^{-1}\|$ (1-norm estimation [11, section 15.3], typically 4 solves).		$4n^2$
Form $B_\theta = B + 2\theta A$, $C_\theta = C + \theta B + \theta^2 A$.		$6n^2$
Cholesky factorizations of B_θ and C_θ.		$2n^3/3$ or less
	Total	$5n^3/3$ or less

To construct the shift θ we use the following strategy: let

$$a = \|A\|, \quad b = \|B\|, \quad c = \|C\|,$$

where $\|\cdot\|$ is any consistent matrix norm. Then, from [18, Lemmas 3.1 and 4.1], for every eigenvalue λ of Q we have

$$|\lambda| \le \frac{1}{2}\|A^{-1}\|\left(b + \sqrt{b^2 + 4c/\|A^{-1}\|}\right) =: \sigma_1, \tag{5.2}$$

$$|\lambda| \le \left(1 + \|A^{-1}\|\right)\max(c^{1/2}, b) =: \sigma_2. \tag{5.3}$$

We take the 1-norm and set $\sigma = \min(\sigma_1, \sigma_2)$. Since σ must greatly overestimate $|\lambda_1|$ when $|\lambda_n| \gg |\lambda_1|$, we carry on one step further and form the shifted quadratic $Q_{-\sigma/2}(\lambda) = Q(\lambda - \sigma/2)$ for which (5.2)–(5.3) give two new bounds τ_1 and τ_2 (and A is unchanged, so $\|A^{-1}\|$ can be reused). We then take $\theta = \min(\sigma, \tau - \frac{1}{2}\sigma)$, where $\tau = \min(\tau_1, \tau_2)$.

As shown by Theorem 3.4, the speed of convergence of iteration (3.7) for overdamped Q depends on the ratio λ_n/λ_{n+1}. An unnecessarily large shift of the spectrum to the left can make this ratio very close to 1, potentially causing slow convergence of the iteration. However, we showed in section 4 that for the generic case of weakly overdamped Q with $\lambda_n = \lambda_{n+1}$ the convergence is at least linear with constant $1/2$, so convergence of the iteration cannot be unduly delayed by a conservative choice of shift.

Table 5.1 details the computations and their cost. (Costs of all the operations used here are summarized in [12, Appendix C].) Preprocessing requires at most $\frac{5}{3}n^3$ flops.

5.2. Overdamping test. The following algorithm is based on Corollary 3.6. It runs the scaled iteration (3.7) until either a non-positive definite B_k or a negative definite $Q(\mu_k)$ is detected, signaling that Q is not overdamped or is overdamped, respectively. The algorithm terminates on one of these conditions or because of possible nonconvergence of the iteration for a non-overdamped Q. It is intended to be applied to Q_θ from the preprocessing step.

ALGORITHM 5.1 (overdamping test). *This algorithm tests whether a quadratic $Q(\lambda) = \lambda^2 A + \lambda B + C$ with $A, B > 0$ and $0 \ne C \ge 0$ is overdamped and, if it is, computes $\mu < 0$ such that $Q(\mu) < 0$. Input parameters are the maximum number of iterations $k_{\max}$ and a convergence tolerance $\epsilon > 0$.*

1 Set $A_0 = A$, $B_0 = B$, $C_0 = C$.
2 Set $\alpha_0 = \|C_0\|_1/\|A_0\|_1$, $\mu_0 = -\alpha_0^{1/2}$, $k = 0$.
3 if $Q(\mu_0) < 0$, $Q(\lambda)$ is (hyperbolic and hence) overdamped, $\mu = \mu_0$, quit, end

4 while $k < k_{\max}$
5 $B_{k+1} = B_k - A_k B_k^{-1} C_k - C_k B_k^{-1} A_k$
6 if $\|B_{k+1} - B_k\|_1 / \|B_{k+1}\|_1 \le \epsilon$, goto line 15, end
7 if $B_{k+1} \not> 0$, Q is not overdamped, quit, end
8 $A_{k+1} = \alpha_k A_k B_k^{-1} A_k$
9 $C_{k+1} = \alpha_k^{-1} C_k B_k^{-1} C_k$
10 $\alpha_{k+1} = \|C_{k+1}\|_1 / \|A_{k+1}\|_1$
11 $\mu_{k+1} = \mu_k \alpha_{k+1}^{1/2^{k+2}}$
12 if $Q(\mu_{k+1}) < 0$, Q is overdamped, $\mu = \mu_{k+1}$, quit, end
13 $k = k + 1$
14 end
15 Q is not overdamped. % See the discussion below.

Note that the crucial definiteness test on line 12 of Algorithm 5.1 is carried out on Q and not on Q_k in (3.5). Hence a positive test can be interpreted irrespective of rounding errors in the iteration: The only errors are in forming $Q(\mu_{k+1})$ and in computing its Cholesky factor. For a non-overdamped Q, it is possible that $B_k > 0$ for all k (see the example at the end of section 4). However, if convergence of the B_k is detected on line 6 then Q is declared not overdamped because by this point an overdamped Q would have been detected, while if $k_{\max}$ is large enough (say, $k_{\max} = 20$) and this iteration limit is reached then Q can reasonably be declared not overdamped in view of the fast (quadratic) convergence of (3.7) for an overdamped Q.

The implementation details of Algorithm 5.1 and the cost per iteration are described in Table 5.2. The total cost for m iterations is $\frac{1}{3}n^3$ flops for $m = 0$ and roughly $\frac{20}{3}mn^3$ flops for $m \ge 1$.

Guo and Lancaster's test for overdamping is based on iteration (3.1), scaled as in (3.7). For the computation of $\widehat{S}$, $\frac{19}{3}\ell n^3$ flops are required, where ℓ is the number of iterations for convergence of (3.1). An extra $5n^3$ flops is needed to form the two solvents $S^{(1)}$ and $S^{(2)}$ (which are nonsymmetric in general) via (3.3). Then the smallest eigenvalue λ_n of $S^{(1)}$ and the largest eigenvalue λ_{n+1} of $S^{(2)}$ need to be computed and the definiteness of $Q((\lambda_n + \lambda_{n+1})/2)$ tested. The total cost is $(\frac{19}{3}\ell + \frac{16}{3})n^3$ flops plus the cost of finding λ_n and λ_{n+1}. Since $m \le \ell$, Algorithm 5.1 is clearly the more efficient, possibly significantly so.

We mention two alternative ways to test hyperbolicity. Both are based on the fact that a Hermitian Q with $A > 0$ is hyperbolic if and only if a certain $2n \times 2n$ pair

TABLE 5.2
Operation count per complete iteration of Algorithm 5.1. *Matrices are assumed real and of dimension* n.

Operations	Cost (flops)
Cholesky factorization of $B_k = L_k L_k^T$ available from previous step.	
Form $V_k = L_k^{-1} A_k$.	n^3
Form $W_k = L_k^{-1} C_k$.	n^3
Compute $A_k B_k^{-1} C_k = V_k^T W_k$.	$2n^3$
Cholesky of B_{k+1}.	$n^3/3$
Compute $A_k B_k^{-1} A_k = V_k^T V_k$.	n^3
Compute $C_k B_k^{-1} C_k = W_k^T W_k$.	n^3
Cholesky of $-Q(\mu_{k+1})$.	$n^3/3$
Total	$20n^3/3$

$(\mathcal{A},\mathcal{B})$ is definite [19, Theorem 3.6]. The first approach is to apply the J-orthogonal Jacobi algorithm of Veselić [28] to $(\mathcal{A},\mathcal{B})$, since the algorithm breaks down when applied to an indefinite pair. Drawbacks of this approach are that the algorithm uses hyperbolic transformations, and so is potentially unstable, and that it must be run to completion to check whether the problem is overdamped, though of course upon completion it has computed the eigenvalues. It requires an initial $\frac{11}{3}n^3$ flops followed by $12sn^3$ flops, where s is the number of sweeps performed. The second approach is to apply to $(\mathcal{A},\mathcal{B})$ an algorithm of Crawford and Moon [4] for detecting definiteness of Hermitian matrix pairs. Although only linearly convergent, this algorithm usually terminates within 30 iterations with a message of "definite," "indefinite," or "fail" (denoting failure of the algorithm to make a determination). The number 30 here is for difficult problems, for which our algorithm may also need 20 iterations. For easy problems, the Crawford–Moon algorithm needs about 3 iterations, while our algorithm needs 0 or 1 iterations. Since the Crawford–Moon algorithm requires one Cholesky factorization per iteration, here of a $2n \times 2n$ matrix, it needs $\frac{8}{3}n^3$ flops per iteration, and this can be reduced to $\frac{1}{3}n^3$ flops per iteration by working directly with the $n \times n$ quadratic Q through the use of a congruence transformation, as given in the proof of [19, Theorem 3.6], for example. Since our algorithm needs $\frac{20}{3}n^3$ flops per iteration, it is often more efficient than the Crawford–Moon algorithm applied to the pair $(\mathcal{A},\mathcal{B})$ and is often less efficient than the Crawford–Moon algorithm working on Q via the congruence. However, the Crawford–Moon algorithm with or without the congruence is numerically unreliable, as we now explain.

We use a MATLAB translation of the Fortran code PDFIND from [3] and also modify it so that it exploits the congruence to work only with the quadratic Q. For the quadratic (3.11), we found that for $\beta \in (0.5196152422, 0.5196152423)$ (which is a small interval in which Q changes from being not overdamped to overdamped—see Tables 3.1 and 3.2) both codes often return with a "fail" message when Algorithm 5.1 correctly diagnoses (non) overdamping. We then considered a scaling of the problem $A \leftarrow \alpha^2 A,\ B \leftarrow \alpha B$, with $\alpha > 0$, which has no effect on the overdamping or on Algorithm 5.1. However, as α decreases, PDFIND becomes more unreliable, due to the increasing ill conditioning of the congruence transformation with decreasing α. To be more specific, we take $\alpha = 10^{-7}$. First, consider $\beta = 0.5157\colon 0.0001\colon 0.5197$. For $\beta = 0.5197$ our algorithm detects overdamping in 3 iterations, and for other values our algorithm detects nonoverdamping in at most 8 iterations. PDFIND, using the congruence, incorrectly detects nonoverdamping for $\beta = 0.5197$ and fails for $\beta = 0.5157$, 0.5177–0.5181, 0.5188, 0.5189, 0.5194. Next, we take $\beta = 0.51965\colon 0.00001\colon 0.51971$. Our algorithm detects overdamping in at most 5 iterations. PDFIND without the congruence incorrectly detects nonoverdamping for 0.51965, 0.51967, 0.51968 and fails for 0.51966, 0.51969, 0.51970. The conclusion is that PDFIND is numerically unreliable whether the congruence is used or not, and when it gives the wrong answer there is no warning. The poor performance of PDFIND when working with the pair is due to the fact that the ill-conditioned congruence transformation is implicitly present in the equivalence between Q being hyperbolic and $(\mathcal{A},\mathcal{B})$ being definite.

For our algorithm, instability could potentially arise if B_k is ill-conditioned. However, we know from [2, p. 40, line 10] that B_k is well-conditioned if $B_0 = B$ is well-conditioned (which is verifiable right from the beginning) and if λ_n/λ_{n+1} is not too close to 1. When λ_n/λ_{n+1} is extremely close to 1, B_k is known to be ill-conditioned for large k. However, B_k appears in our algorithm only in terms like $A_k B_k^{-1} C_k$, and A_k and C_k converge to 0, so the ill-conditioning of B_k has only a limited effect on our algorithm. Indeed, instability has not been observed in any of our tests.

TABLE 5.3
Operation count for the eigenvalue computation, with reference to (5.4).

Operations	Cost (flops)
Cholesky factorizations of $A = L_A L_A^T$ and $-C = L_C L_C^T$ already available.	
Form $R = -(L_A^{-1} L_C)^T$.	$n^3/3$
Form $G = L_A^{-1} B L_A^{-T}$.	$3n^3/2$
Tridiagonalization of $\left[\begin{smallmatrix} -G & R \\ -R^T & 0 \end{smallmatrix}\right]$.	$< 4(2n)^3/3$
Eigenvalues via (e.g.) QR iteration.	$O(n^2)$
Total	$\approx 13n^3$

5.3. Solving hyperbolic QEPs via definite linearizations. Recall that the scalar μ computed by Algorithm 5.1 applied to Q_θ is such that $Q(\mu+\theta) = Q_\theta(\mu) < 0$. Hence with $\omega = \mu + \theta$ we have

$$\begin{aligned}\widetilde{Q}(t) = Q(t+\omega) &= t^2 A + t(B + 2\omega A) + C + \omega B + \omega^2 A \\ &= t^2 \widetilde{A} + t\widetilde{B} + \widetilde{C},\end{aligned}$$

with $\widetilde{C} = Q(\omega) < 0$ and $\widetilde{A} = A > 0$. The pencils

$$L_1(\lambda) = \lambda \begin{bmatrix} \widetilde{A} & 0 \\ 0 & -\widetilde{C} \end{bmatrix} + \begin{bmatrix} \widetilde{B} & \widetilde{C} \\ \widetilde{C} & 0 \end{bmatrix}, \quad L_2(\lambda) = \lambda \begin{bmatrix} 0 & \widetilde{A} \\ \widetilde{A} & \widetilde{B} \end{bmatrix} + \begin{bmatrix} -\widetilde{A} & 0 \\ 0 & \widetilde{C} \end{bmatrix}$$

are both Hermitian definite linearizations of $\widetilde{Q}$ with a positive definite leading coefficient of L_1 and a negative definite trailing coefficient of L_2. They share the same eigenvalues as $\widetilde{Q}$, and the eigenvectors of $\widetilde{Q}$ are easy to recover from those of L_1 or L_2. The sensitivity and stability of these linearizations have recently been studied in [14], [15], [17]. It is shown therein that the scaling of Fan, Lin and Van Dooren [6] should be applied to $\widetilde{Q}$ before linearizing. The choice between L_1 and L_2 should be guided by the fact that, in terms of conditioning and backward error, they favor large and small eigenvalues, respectively. However, if $\widetilde{C}$ or $\widetilde{A}$ is well-conditioned and $\|\widetilde{B}\|/(\|\widetilde{A}\|\,\|\widetilde{C}\|)^{1/2}$ is not much bigger than 1 then L_1 or L_2, respectively, can safely be used to stably obtain all of the eigenpairs. For more details on conditioning and backward error for L_1 and L_2, see [14], [15], [17].

Using Cholesky factorizations $\widetilde{A} = L_A L_A^T$ and $-\widetilde{C} = L_C L_C^T$, the definite generalized eigenvalue problem $L_1(\lambda)z = 0$ or $L_2(\lambda)z = 0$ is transformed to a Hermitian (or real symmetric) standard eigenvalue problem [5]. For example, $L_1(\lambda)$ reduces to

$$\lambda I + \begin{bmatrix} L_A^{-1} \widetilde{B} L_A^{-T} & -L_A^{-1} L_C \\ -L_C^T L_A^{-T} & 0 \end{bmatrix}. \tag{5.4}$$

As Table 5.3 explains, this phase requires about $13n^3$ flops, giving a grand total of $(\frac{20}{3}m + 13)n^3$ flops.

Guo and Lancaster's solution algorithm has a total cost of $(\frac{19}{3}\ell + 25)n^3$ flops, assuming the eigenvalues of $S^{(1)}$ and $S^{(2)}$ (which are the eigenvalues of Q) are computed by the QR algorithm. In practice this is significantly more than the cost of our algorithm given that $m \le \ell$ is usually small.

The most common way of solving the QEP is to apply the QZ algorithm or a Krylov method to a linearization L of Q. The QZ algorithm applied to the $2n \times 2n$ L costs $240n^3$ flops for the computation of the eigenvalues.

Our algorithm has two important advantages over that of Guo and Lancaster and QZ applied to a linearization, besides its more favorable operation count. First, it works entirely with symmetric matrices, which reduces the storage requirement. Second, it guarantees to produce real eigenvalues in floating point arithmetic; the other two approaches cannot do so because they invoke the QZ algorithm and the nonsymmetric QR algorithm.

6. Numerical experiment. We describe an experiment that illustrates the behavior of our algorithm for testing overdamping. More extensive testing of this algorithm, and of the preprocessing and solving procedures described in section 5, will be presented in a future publication. Our experiments were performed in MATLAB 7.4 (R2007a), for which the unit roundoff is $u = 2^{-53} \approx 1.1 \times 10^{-16}$. We took $k_{\max} = 30$ and $\epsilon = u$ in Algorithm 5.1.

We first describe a useful technique for generating symmetric quadratic matrix polynomials with prescribed eigenvalues and eigenvectors and positive definite coefficient matrices.

Let (λ_k, v_k), $k = 1\colon 2n$, be a set of given real eigenpairs such that, with

$$\begin{aligned} \Lambda &= \operatorname{diag}(\lambda_1, \dots, \lambda_{2n}) =: \Lambda_1 \oplus \Lambda_2, && \Lambda_1, \Lambda_2 \in \mathbb{R}^{n\times n}, \\ V &:= [\, v_1, \dots, v_{2n} \,] =: [\, V_1 \quad V_2 \,], && V_1, V_2 \in \mathbb{R}^{n\times n}, \end{aligned}$$

V_1 and V_2 are nonsingular, and

$$V_1 V_1^T = V_2 V_2^T, \qquad V_1 \Lambda_1 V_1^T - V_2 \Lambda_2 V_2^T =: \Gamma \text{ is nonsingular.} \tag{6.1}$$

Then the symmetric quadratic polynomial defined by the matrices

$$A = \Gamma^{-1}, \quad B = -A(V_1 \Lambda_1^2 V_1^T - V_2 \Lambda_2^2 V_2^T)A, \tag{6.2a}$$

$$C = -A(V_1 \Lambda_1^3 V_1^T - V_2 \Lambda_2^3 V_2^T)A + B\Gamma B \tag{6.2b}$$

has eigenpairs (λ_k, v_k), $k = 1\colon 2n$ (see [23] for example). We now show how to generate a potentially overdamped quadratic.

LEMMA 6.1. *Assume that* $0 > \lambda_1 \geq \cdots \geq \lambda_n > \lambda_{n+1} \geq \cdots \geq \lambda_{2n}$. *Then* Γ *is nonsingular and the matrices generated by* (6.2) *satisfy* $A > 0$, $B > 0$, *and* $C > 0$.

Proof. It follows from Weyl's theorem [20, p. 181] that $\Gamma > 0$ and hence that $A > 0$. All matrices V_2 that satisfy the first constraint in (6.1) can be written as $V_1 U$ for some orthogonal U. Hence we can write

$$B = -AV_1(\Lambda_1^2 - U\Lambda_2^2 U^T)V_1^T A = -AV_1(H_1^2 - H_2^2)V_1^T A,$$

where $H_1 = \Lambda_1$ and $H_2 = U\Lambda_2 U^T$, and again Weyl's theorem guarantees that $B > 0$.

It is known that (V, Λ, PV^T), where $P = \operatorname{diag}(I_n, -I_n)$, forms a self-adjoint triple for $Q(\lambda)$ [7, section 10.2]. Since Q has no zero eigenvalues, C is nonsingular and a formula for its inverse is easily obtained from the resolvent form of $Q(\lambda)$: For $\lambda \neq \lambda_i$,

$$Q(\lambda)^{-1} = V(\lambda I_{2n} - \Lambda)^{-1} PV^T.$$

TABLE 6.1
Minimum, average, and maximum number of iterations performed by Algorithm 5.1 *and percentage of overdamped problems, for each* n *and matrix type.*

n	type 1		type 2		type 3	
5	0.0, 2.4, 6.0	100%	0.0, 0.8, 3.0	100%	0.0, 2.4, 5.0	25%
10	0.0, 3.6, 10.0	100%	0.0, 0.5, 3.0	100%	2.0, 2.7, 4.0	5%
50	0.0, 4.2, 11.0	100%	0.0, 2.1, 4.0	100%	2.0, 2.1, 3.0	0%
100	3.0, 6.2, 10.0	100%	0.0, 2.6, 4.0	100%	2.0, 2.0, 2.0	0%
250	2.0, 6.0, 11.0	100%	2.0, 3.0, 4.0	100%	2.0, 2.0, 2.0	0%
500	3.0, 7.5, 11.0	100%	2.0, 3.0, 4.0	100%	2.0, 2.0, 2.0	0%

Setting $\lambda = 0$ in the above expression gives

$$C^{-1} = -V\Lambda^{-1}PV^T = -V_1(H_1^{-1} - H_2^{-1})V_1^T,$$

and once again Weyl's theorem guarantees that C^{-1}, and therefore also C, is positive definite. ∎

We use the following eigenvalue distributions:

type 1: λ_k, $k = 1\colon 2n$, is uniformly distributed in $[-100, -1]$.

type 2: λ_k is uniformly distributed in $[-100, -6]$ for $k = n+1\colon 2n$ and $[-5, -1]$ for $k = 1\colon n$.

type 3: λ_k is uniformly distributed in $[-100, 20]$. B and C are then shifted as in (2.2) with $\theta = 1.1\lambda_1$ to ensure that the eigenvalues are all negative.

We took $V_1 = U_1$ and $V_2 = V_1U_2$, where U_1 and U_2 are random orthogonal matrices from the Haar distribution [11, section 28.3]. For types 1 and 2, A, B, and C are all positive definite by construction; for type 3 nothing can be said about the definiteness of A, B, and C. Table 6.1 shows the minimum, average, and maximum number of iterations for Algorithm 5.1 over 20 quadratics for each of several values of n, along with the percentage of Q found to be overdamped for each n and matrix type. In all cases where Q was deemed overdamped, the computed μ was verified to lie in $(\lambda_{n+1}, \lambda_n)$.

We make several observations:

- For all three eigenvalue distributions, Algorithm 5.1 is quick to terminate, especially for types 2 and 3, with only very occasional need for more than 10 iterations. The gap between λ_n and λ_{n+1} is larger for type 2 than type 1, which explains the greater number of iterations for type 1.
- With V_1 orthogonal the coefficients matrices A, B, and C are well-conditioned, with 2-norm condition numbers of order 10^2. If instead we take V_1 as a random matrix with 2-norm condition number 10^4 (computed in MATLAB as `gallery('randsvd',n,1e4,...)`, the condition numbers of A, B, and C are of order 10^8 and the number of iterations of the algorithm increases, though only slightly: The maximum number of iterations over all tests is 13, and the largest average over all n rises to 7.8, 3.1, and 3.2 for types 1, 2, and 3, respectively.
- After detecting overdamping an average of 6–9 more iterations are needed for convergence of the block cyclic iteration. Recall that the algorithm of Guo and Lancaster [9] needs to iterate to convergence in order to show overdamping.

Acknowledgments. This work was started while the first author visited MIMS in the School of Mathematics at the University of Manchester in 2005; he thanks the School for its hospitality. We thank Qiang Ye for helpful comments concerning the algorithm of Crawford and Moon.

REFERENCES

[1] L. Barkwell and P. Lancaster, *Overdamped and gyroscopic vibrating systems*, Trans. ASME J. Appl. Mech., 59 (1992), pp. 176–181.

[2] D. A. Bini, L. Gemignani, and B. Meini, *Computations with infinite Toeplitz matrices and polynomials*, Linear Algebra Appl., 343–344 (2002), pp. 21–61.

[3] C. R. Crawford, *ALGORITHM* 646 *PDFIND: A routine to find a positive definite linear combination of two real symmetric matrices*, ACM Trans. Math. Software, 12 (1986), pp. 278–282.

[4] C. R. Crawford and Y. S. Moon, *Finding a positive definite linear combination of two Hermitian matrices*, Linear Algebra Appl., 51 (1983), pp. 37–48.

[5] P. I. Davies, N. J. Higham, and F. Tisseur, *Analysis of the Cholesky method with iterative refinement for solving the symmetric definite generalized eigenproblem*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 472–493.

[6] H.-Y. Fan, W.-W. Lin, and P. Van Dooren, *Normwise scaling of second order polynomial matrices*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 252–256.

[7] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Academic Press, New York, 1982.

[8] C.-H. Guo, *Convergence rate of an iterative method for a nonlinear matrix equation*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 295–302.

[9] C.-H. Guo and P. Lancaster, *Algorithms for hyperbolic quadratic eigenvalue problems*, Math. Comp., 74 (2005), pp. 1777–1791.

[10] N. J. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.

[11] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, PA, 2002.

[12] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, PA, 2008.

[13] N. J. Higham and H.-M. Kim, *Numerical analysis of a quadratic matrix equation*, IMA J. Numer. Anal., 20 (2000), pp. 499–519.

[14] N. J. Higham, R.-C. Li, and F. Tisseur, *Backward error of polynomial eigenproblems solved by linearization*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1218–1241.

[15] N. J. Higham, D. S. Mackey, and F. Tisseur, *The conditioning of linearizations of matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1005–1028.

[16] N. J. Higham, D. S. Mackey, and F. Tisseur, *Definite matrix polynomials and their linearization by definite pencils*, SIAM J. Matrix Anal. Appl., to appear.

[17] N. J. Higham, D. S. Mackey, F. Tisseur, and S. D. Garvey, *Scaling, sensitivity and stability in the numerical solution of quadratic eigenvalue problems*, Internat. J. Numer. Methods Engrg., 73 (2008), pp. 344–360.

[18] N. J. Higham and F. Tisseur, *Bounds for eigenvalues of matrix polynomials*, Linear Algebra Appl., 358 (2003), pp. 5–22.

[19] N. J. Higham, F. Tisseur, and P. M. Van Dooren, *Detecting a definite Hermitian pair and a hyperbolic or elliptic quadratic eigenvalue problem, and associated nearness problems*, Linear Algebra Appl., 351–352 (2002), pp. 455–474.

[20] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[21] D. J. Inman and A. N. Andry, Jr., *Some results on the nature of eigenvalues of discrete damped linear systems*, Trans. ASME J. Appl. Mech., 47 (1980), pp. 927–930.

[22] P. Lancaster, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, 1966. Reprinted by Dover, New York, 2002.

[23] P. Lancaster, *Inverse spectral problems for semisimple damped vibrating systems*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 279–301.

[24] W.-W. Lin and S.-F. Xu, *Convergence analysis of structure-preserving doubling algorithms for Riccati-type matrix equations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 26–39.

[25] A. S. Markus, *Introduction to the Spectral Theory of Polynomial Operator Pencils*, American Mathematical Society, Providence, RI, 1988.

[26] B. Meini, *Efficient computation of the extreme solutions of* $X + A^*X^{-1}A = Q$ *and* $X - A^*X^{-1}A = Q$, Math. Comp., 71 (2002), pp. 1189–1204.

[27] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

[28] K. Veselić, *A Jacobi eigenreduction algorithm for definite matrix pairs*, Numer. Math., 64 (1993), pp. 241–269.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1614–1638

A METHOD TO AVOID DIVERGING COMPONENTS IN THE CANDECOMP/PARAFAC MODEL FOR GENERIC $I \times J \times 2$ ARRAYS*

ALWIN STEGEMAN† AND LIEVEN DE LATHAUWER‡

Abstract. Computing the Candecomp/Parafac (CP) solution of R components (i.e., the best rank-R approximation) for a generic $I \times J \times 2$ array may result in diverging components, also known as "degeneracy." In such a case, several components are highly correlated in all three modes, and their component weights become arbitrarily large. Evidence exists that this is caused by the nonexistence of an optimal CP solution. Instead of using CP, we propose to compute the best approximation by means of a generalized Schur decomposition (GSD), which always exists. The obtained GSD solution is the limit point of the sequence of CP updates (whether it features diverging components or not) and can be separated into a nondiverging CP part and a sparse Tucker3 part or into a nondiverging CP part and a smaller GSD part. We show how to obtain both representations and illustrate our results with numerical experiments.

Key words. canonical decomposition, parallel factors analysis, low-rank tensor approximations, degenerate Parafac solutions, diverging components

AMS subject classifications. 15A18, 15A22, 15A69, 49M27, 62H25

DOI. 10.1137/070692121

1. Introduction.

Hitchcock [16, 17] introduced a generalized rank and related decomposition of a multiway array or tensor. The same decomposition was proposed independently by Carroll and Chang [3] and Harshman [13] for component analysis of three-way data arrays. They named it Candecomp and Parafac, respectively. We denote the Candecomp/Parafac (CP) model, i.e., the decomposition with a residual term, as

$$\underline{\mathbf{Z}} = \sum_{h=1}^{R} \omega_h \, (\mathbf{a}_h \otimes \mathbf{b}_h \otimes \mathbf{c}_h) + \underline{\mathbf{E}}, \tag{1.1}$$

where $\underline{\mathbf{Z}}$ is an $I \times J \times K$ data array, ω_h is the weight of component h, $\otimes$ denotes the outer product, and $\|\mathbf{a}_h\| = \|\mathbf{b}_h\| = \|\mathbf{c}_h\| = 1$ for $h = 1, \ldots, R$, with $\|\cdot\|$ denoting the Frobenius norm. To find the R components $\mathbf{a}_h \otimes \mathbf{b}_h \otimes \mathbf{c}_h$ and the weights ω_h, an iterative algorithm is used which minimizes the Frobenius norm of the residual

*Received by the editors May 16, 2007; accepted for publication (in revised form) by N. Mastronardi September 22, 2008; published electronically January 16, 2009.
http://www.siam.org/journals/simax/30-4/69212.html

†Corresponding author. Heymans Institute for Psychological Research, University of Groningen, Grote Kruisstraat 2/1, 9712 TS Groningen, The Netherlands (a.w.stegeman@rug.nl, http://www.gmw.rug.nl/∼stegeman). This author's research was supported by the Dutch Organization for Scientific Research (NWO), VENI grant 451-04-102.

‡Subfaculty Science and Technology, Katholieke Universiteit Leuven, Campus Kortrijk, E. Sabbelaan 53, 8500 Kortrijk, Belgium (Lieven.DeLathauwer@kuleuven-kortrijk.be) and Research Division SCD, Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (Lieven.DeLathauwer@esat.kuleuven.be, http://homes.esat.kuleuven.be/∼delathau/home.html). This author's research was supported by (1) Research Council K.U.Leuven: GOA-Ambiorics, CoE EF/05/006 Optimization in Engineering (OPTEC), CIF1, STRT1/08/23; (2) F.W.O. (a) project G.0321.06; (b) Research Communities ICCoS, ANMMM, and MLDM; (3) the Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, "Dynamical systems, control, and optimization," 2007–2011); (4) EU: ERNSI. A large part of this research was carried out when the author was with the French Centre National de la Recherche Scientifique (C.N.R.S.).

array $\underline{\mathbf{E}}$. For an overview and comparison of CP algorithms, see Hopke et al. [18] and Tomasi and Bro [45].

The rank of a three-way array $\underline{\mathbf{Z}}$ is defined in the usual way, i.e., the smallest number of rank-1 arrays whose sum equals $\underline{\mathbf{Z}}$. A three-way array has rank 1 if it is the outer product of three vectors, i.e., $\mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}$. We denote three-way rank as $\text{rank}_{\otimes}(\underline{\mathbf{Z}})$. It follows that the CP model tries to find a best rank-R approximation to the three-way array $\underline{\mathbf{Z}}$.

The real-valued CP model, i.e., where $\underline{\mathbf{Z}}$ and the model parameters are real-valued, was introduced in psychometrics (Carroll and Chang [3]) and phonetics (Harshman [13]). Later on, it was also applied in chemometrics and the food industry (Bro [1] and Smilde, Bro, and Geladi [37]). For other applications of CP in psychometrics, see Kroonenberg [25]. Complex-valued applications of CP occur in signal processing, especially wireless telecommunications; see Sidiropoulos, Giannakis, and Bro [35], Sidiropoulos, Bro, and Giannakis [36], and De Lathauwer and Castaing [9]. Also, CP describes the basic structure of fourth-order cumulants of multivariate data on which a lot of algebraic methods for independent component analysis are based (Comon [4], De Lathauwer, De Moor, and Vandewalle [5], and Hyvärinen, Karhunen, and Oja [20]). In this paper, we consider the real-valued CP model. All occurrences of three-way rank are assumed to be over the real field.

For later use, we mention that the CP model (1.1) is a special case of the Tucker3 model of Tucker [46]. The latter is defined as

$$\underline{\mathbf{Z}} = \sum_{h=1}^{R} \sum_{i=1}^{P} \sum_{j=1}^{Q} g_{hij} \, (\mathbf{a}_h \otimes \mathbf{b}_i \otimes \mathbf{c}_j) + \underline{\mathbf{E}}. \tag{1.2}$$

Clearly, the case with $R = P = Q$ and $g_{hij} = 0$ if $(h, i, j) \neq (h, h, h)$ yields (1.1). The $R \times P \times Q$ array $\underline{\mathbf{G}}$ with entries g_{hij} is referred to as the core array. The matrices $[\mathbf{a}_1| \ldots |\mathbf{a}_R]$, $[\mathbf{b}_1| \ldots |\mathbf{b}_P]$, and $[\mathbf{c}_1| \ldots |\mathbf{c}_Q]$ are called the component matrices.

A matrix notation of the CP model (1.1) is as follows. Let $\mathbf{Z}_k$ $(I \times J)$ and $\mathbf{E}_k$ $(I \times J)$ denote the kth slices of $\underline{\mathbf{Z}}$ and $\underline{\mathbf{E}}$, respectively. Then (1.1) can be written as

$$\mathbf{Z}_k = \mathbf{A}\, \mathbf{C}_k\, \mathbf{B}^T + \mathbf{E}_k, \qquad k = 1, \ldots K, \tag{1.3}$$

where the component matrices $\mathbf{A}$ $(I \times R)$ and $\mathbf{B}$ $(J \times R)$ have the vectors $\mathbf{a}_h$ and $\mathbf{b}_h$ as columns, respectively, and $\mathbf{C}_k$ $(R \times R)$ is the diagonal matrix with the kth elements of the vectors $\omega_h \mathbf{c}_h$ on its diagonal. The model part of the CP model is characterized by $(\mathbf{A}, \mathbf{B}, \mathbf{C})$, where component matrix $\mathbf{C}$ $(K \times R)$ has the vectors $\mathbf{c}_h$ as columns. Hence, it is assumed that the weights ω_h are absorbed by the matrix $\mathbf{C}$.

The most attractive feature of CP is its uniqueness property. Kruskal [26] has shown that, for fixed residuals $\underline{\mathbf{E}}$, the vectors $\mathbf{a}_h$, $\mathbf{b}_h$, and $\mathbf{c}_h$ and the weights ω_h are unique up to sign changes and a reordering of the summands in (1.1) if

$$k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geq 2\, R + 2, \tag{1.4}$$

where $k_{\mathbf{A}}$, $k_{\mathbf{B}}$, $k_{\mathbf{C}}$ denote the k-ranks of the component matrices. The k-rank of a matrix is the largest number x such that every subset of x columns of the matrix is linearly independent. If a CP solution is unique up to these indeterminacies, it is called *essentially unique*. Two CP solutions which are identical up to the essential uniqueness indeterminacies will be called *equivalent*.

In case one of the component matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ has full column rank, a weaker uniqueness condition than (1.4) has been derived by Jiang and Sidiropoulos [22] and De Lathauwer [7]. See also Stegeman, Ten Berge, and De Lathauwer [41].

The practical use of CP has been hampered by the occurrence of diverging CP components, also known as "degeneracy." In such cases, convergence of a CP algorithm is extremely slow, and some components display the following pattern. Let the model parameters of the nth update of a CP algorithm be denoted by a superscript (n). For the diverging components, the weights $\omega_h^{(n)}$ become arbitrarily large in magnitude, and the corresponding columns in $\mathbf{A}^{(n)}$, $\mathbf{B}^{(n)}$, and $\mathbf{C}^{(n)}$ become nearly linearly dependent. Although the individual diverging components may diverge in nearly opposite directions, their sum still contributes to a better fit of the CP model. Diverging CP components are a problem to the analysis of three-way arrays, since the obtained CP solution is hardly interpretable. The occurrence of diverging components can be avoided by imposing orthogonality constraints on the components matrices; see Krijnen, Dijkstra and Stegeman [24], but this will come with some loss of fit. Lim [29] shows that diverging components do not occur for nonnegative $\underline{\mathbf{Z}}$ under the restriction of nonnegative component matrices.

The first case of diverging CP components was reported in Harshman and Lundy [14]. Contrived examples are given by Ten Berge, Kiers, and De Leeuw [43] and Paatero [33]. Kruskal, Harshman, and Lundy [27] have argued that diverging CP components occur due to the fact that the array $\underline{\mathbf{Z}}$ has no best rank-R approximation, i.e., CP has no optimal solution. They reason that every sequence of CP updates, of which the objective value is approaching the infimum of the CP objective function, must fail to converge and displays a pattern of a diverging CP components. This has recently been proven by Krijnen, Dijkstra, and Stegeman [24].

De Silva and Lim [10] give results on the existence of a best rank-R approximation of N-way arrays with $N \geq 3$. For the three-way CP model, [10] shows that for $R = 1$, an optimal CP solution always exists, while for any $I, J, K \geq 2$ and any $R \in \{2, \ldots, \min(I, J, K)\}$, a rank-$(R+1)$ array $\underline{\mathbf{Z}}$ exists which has no optimal CP solution. Also, [10] shows that all $2 \times 2 \times 2$ arrays of rank 3 (a set of positive volume in $\mathbb{R}^{2\times 2\times 2}$) have no optimal CP solution for $R = 2$ and that, for any $I, J, K \geq 2$, the set of arrays in $\mathbb{R}^{I\times J\times K}$, which have no optimal CP solution for $R = 2$ has positive volume.

Stegeman [38, 40] has mathematically analyzed diverging CP components occurring for generic $I \times J \times 2$ arrays $\underline{\mathbf{Z}}$ and all values of R. In these cases, diverging components occur if the sequence of CP updates converges to a limit point $\underline{\mathbf{X}}$, which has rank larger than R. Formally, these occurrences of diverging components can be described as follows. There exist disjoint index sets $D_1, \ldots, D_r \subset \{1, \ldots, R\}$ such that

$$|\omega_h^{(n)}| \to \infty, \qquad \text{for all } h \in D_j, \quad j = 1, \ldots, r, \tag{1.5}$$

$$\text{while} \quad \left\| \sum_{h \in D_j} \omega_h^{(n)} \left(\mathbf{a}_h^{(n)} \otimes \mathbf{b}_h^{(n)} \otimes \mathbf{c}_h^{(n)}\right) \right\| \qquad \text{is bounded}, \quad j = 1, \ldots, r. \tag{1.6}$$

Stegeman [38, 40] gives a complete characterization of the diverging components (1.5)–(1.6) in terms of properties of the limit point of the sequence of CP updates. Also, [40] provides a link between diverging CP components and results from the theory of matrix pencils and algebraic complexity theory.

The only mathematically analyzed cases of diverging CP components so far are the contrived examples in Ten Berge, Kiers, and De Leeuw [43] and Paatero [33], generic $I \times J \times 2$ arrays in Stegeman [38, 40], and generic $5 \times 3 \times 3$ and $8 \times 4 \times 3$ arrays, and generic $3 \times 3 \times 4$ and $3 \times 3 \times 5$ arrays with symmetric slices in Stegeman [39].

A numerical example of diverging CP components is the following. Let $\underline{\mathbf{Z}}$ be a $4 \times 4 \times 2$ array with slices

$$\mathbf{Z}_1 = \begin{bmatrix} -0.5 & -1.2 & 0.3 & -0.6 \\ -1.7 & 1.1 & 0.1 & 2.1 \\ 0.1 & 1.1 & -0.2 & -0.2 \\ 0.2 & -0.1 & 0.7 & 0.1 \end{bmatrix} \quad \text{and} \quad \mathbf{Z}_2 = \begin{bmatrix} 0.8 & 1.1 & -1.7 & -0.9 \\ 0.7 & -1.3 & 0.2 & 0.5 \\ 1.2 & -0.1 & -1.1 & 0.2 \\ 0.6 & -0.2 & 1.4 & -1.0 \end{bmatrix}. \tag{1.7}$$

This array was randomly generated such that $\text{rank}_{\otimes}(\underline{\mathbf{Z}}) = 5$. Next, we try to fit the CP model with $R = 4$ components using the multilinear engine of Paatero [32]. For a convergence criterion of `1e-15`, the algorithm terminates after 162055 iterations with an objective value of 0.051204 and final CP update

$$\mathbf{A} = \begin{bmatrix} 0.6787 & 0.1278 & 0.6767 & -0.6778 \\ -0.6642 & -0.7946 & -0.6735 & 0.6693 \\ -0.1189 & -0.5895 & -0.1464 & 0.1320 \\ -0.2898 & 0.0690 & -0.2590 & 0.2746 \end{bmatrix}, \tag{1.8}$$

$$\mathbf{B} = \begin{bmatrix} -0.6870 & -0.8259 & -0.6919 & -0.6895 \\ -0.2365 & -0.0386 & -0.2609 & -0.2481 \\ -0.0509 & 0.4005 & -0.0080 & -0.0298 \\ 0.6852 & 0.3949 & 0.6732 & 0.6800 \end{bmatrix}, \tag{1.9}$$

$$\mathbf{C} = \begin{bmatrix} 1454 & -2.8913 & 1443 & 2895 \\ 789 & 4.4617 & 634 & 1426 \end{bmatrix}, \tag{1.10}$$

where the columns of $\mathbf{A}$ and $\mathbf{B}$ are normalized to length 1. It can be seen that columns 1, 3, and 4 in $\mathbf{A}$ and $\mathbf{B}$ are nearly identical up to a sign change. Also, these columns have large magnitudes in $\mathbf{C}$. Hence, CP components 1, 3, and 4 appear to be diverging. The multilinear engine terminates with nearly the same CP update for all tried random starting values. The alternating least squares CP algorithm gives the same results.

Since diverging CP components cannot be interpreted, one may wonder whether they can be avoided. However, the discussion above shows that for some array sizes and some values of R, there is no best rank-R approximation and, hence, trying to fit the CP model results in diverging components. To ensure the existence of a best rank-R approximation, De Silva and Lim [10] propose to consider the closure of the set of arrays with at most rank R instead. For each array size and value of R, this involves characterizing the boundary arrays of this set. These are the limit points of the sequences of CP updates featuring diverging CP components. De Silva and Lim [10] show that for $R = 2$, these limit points have rank 3 with the following decomposition into rank-1 terms:

$$\underline{\mathbf{X}} = \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{y}_3 + \mathbf{x}_1 \otimes \mathbf{y}_2 \otimes \mathbf{x}_3 + \mathbf{y}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3. \tag{1.11}$$

In this paper, we apply the idea of De Silva and Lim [10] to the CP model for generic $I \times J \times 2$ arrays $\underline{\mathbf{Z}}$. Apart from the results in Stegeman [39], this is the only class of arrays for which the analysis of diverging components is nearly complete. Instead of fitting the CP model, we propose to find the best approximation of $\underline{\mathbf{Z}}$ in terms of the generalized Schur decomposition (GSD), which was considered in De Lathauwer, De Moor, and Vandewalle [6]. The GSD model is the same as (1.3) except that $\mathbf{A}$ and

$\mathbf{B}$ are columnwise orthonormal and $\mathbf{C}_k$ are upper triangular $k = 1, 2$. We show that an optimal solution to the GSD model always exists. Moreover, for $I \times J \times 2$ arrays, the set of feasible GSD solutions equals the closure of the set of feasible CP solutions. Hence, the optimal GSD solution, if it is unique, is the limit point of the sequence of CP updates, whether the latter features diverging components or not.

Next, we show how to write the obtained GSD solution in several alternative forms. First, using the Jordan normal form, the GSD solution may be written as the sum of the nondiverging CP components and a sparse Tucker3 part. Here, each of the m sets of diverging CP components in (1.5)–(1.6) forms one block in the Tucker3 part. We call this the CP+Jordan form. Although this is not a decomposition into rank-1 terms, it is an essentially unique decomposition, and its blocks may be interpretable to the researcher. Second, the obtained GSD solution may be written as the sum of the nondiverging CP components and a smaller GSD part. We call this the CP+GSD form. If one is only interested in obtaining the nondiverging CP components, this is a fast way to get them. Third, using the CP+Jordan form, the GSD solution may also be written as a sum of rank-1 terms where the number of terms equals the rank of the solution array. However, this rank-revealing decomposition is not essentially unique. During the computation of the GSD solution, the problems of diverging CP components do not arise, neither during the computation of the mentioned alternative forms for the GSD solution.

As explained above, the analyzed cases of diverging CP components most likely occur because the CP model has no optimal solution. Hence, modified CP algorithms designed to avoid diverging components (e.g., Rayens and Mitchell [34], Cao et al. [2]) are no remedy here. With our method for $I \times J \times 2$ arrays, the problems of diverging CP components are avoided without imposing additional constraints.

Note that the occurrences of diverging CP components we consider do not include cases where $\text{rank}_{\otimes}(\underline{\mathbf{Z}}) \leq R$ and either its full CP decomposition resembles a case of diverging components or where diverging components occur due to an unlucky choice of the starting position of the CP algorithm. Examples of these cases can be found in Mitchell and Burdick [30] and Paatero [33]. We will assume instead that $\text{rank}_{\otimes}(\underline{\mathbf{Z}}) > R$.

This paper is organized as follows. We discuss the analysis of diverging CP components for typical $I \times I \times 2$ arrays $\underline{\mathbf{Z}}$ of rank $I + 1$ and $R = I$ in section 3. For this, we need results on the rank of $I \times I \times 2$ arrays. These are presented in section 2. In section 4 we discuss the simultaneous GSD model. In section 5, we consider the GSD model for $I \times I \times 2$ arrays and show how it is related to the CP model. In section 6, we show how to obtain the CP+Jordan and CP+GSD representations of the GSD solution. In section 7, we discuss the extension of our analysis for $I \times I \times 2$ arrays and $R = I$ to $I \times J \times 2$ arrays and general R. Section 8 contains numerical experiments which illustrate our results. Finally, section 9 provides a discussion.

2. The rank of $I \times I \times 2$ arrays. For an array $\underline{\mathbf{Y}} \in \mathbb{R}^{I \times I \times 2}$, we denote its $I \times I$ frontal slices by $\mathbf{Y}_k$, $k = 1, 2$. Let

$$\mathcal{R}_I = \{\underline{\mathbf{Y}} \in \mathbb{R}^{I \times I \times 2} : \det(\mathbf{Y}_k) \neq 0\,,\ k = 1, 2\}. \tag{2.1}$$

The following result on the rank of arrays in $\mathcal{R}_I$ is due to Ja' Ja' [21]. For later use, we also give its proof as formulated in Stegeman [38].

LEMMA 2.1. *For $\underline{\mathbf{X}} \in \mathcal{R}_I$, the following statements hold:*

(i) *If $\mathbf{X}_2\mathbf{X}_1^{-1}$ has I real eigenvalues and is diagonalizable, then $\underline{\mathbf{X}}$ has rank I.*

(ii) *If* $\mathbf{X}_2\mathbf{X}_1^{-1}$ *has at least one pair of complex eigenvalues, then* $\underline{\mathbf{X}}$ *has at least rank* $I+1$.

(iii) *If* $\mathbf{X}_2\mathbf{X}_1^{-1}$ *has* I *real eigenvalues but is not diagonalizable, then* $\underline{\mathbf{X}}$ *has at least rank* $I+1$.

Proof. If (i) holds, then $\mathbf{X}_2\mathbf{X}_1^{-1}$ has an eigendecomposition $\mathbf{K}\,\mathbf{\Lambda}\,\mathbf{K}^{-1}$, where $\mathbf{\Lambda}$ is the $I \times I$ diagonal matrix of eigenvalues and $\mathbf{K}$ contains the associated eigenvectors. Taking

$$\mathbf{A} = \mathbf{K}, \qquad \mathbf{B}^T = \mathbf{K}^{-1}\,\mathbf{X}_1, \qquad \mathbf{C}_1 = \mathbf{I}_I, \qquad \mathbf{C}_2 = \mathbf{\Lambda}, \tag{2.2}$$

yields a full rank-I decomposition of $\underline{\mathbf{X}}$ as in (1.3).

The proof of (ii)–(iii) is as follows. Since its $I \times I$ slices are nonsingular, it follows that $\underline{\mathbf{X}}$ has at least rank I. Suppose $\underline{\mathbf{X}}$ has rank I. Then there exist nonsingular matrices $\mathbf{A}$ and $\mathbf{B}$ and nonsingular diagonal matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ such that $\mathbf{X}_k = \mathbf{A}\,\mathbf{C}_k\,\mathbf{B}^T$, $k = 1, 2$. But then $\mathbf{X}_2\mathbf{X}_1^{-1} = \mathbf{A}\,\mathbf{C}_2\,\mathbf{C}_1^{-1}\,\mathbf{A}^{-1}$ is an eigendecomposition with I real eigenvalues and I linearly independent eigenvectors, which contradicts (ii)–(iii). Hence, the rank of $\underline{\mathbf{X}}$ is at least $I+1$. ∎

If $\underline{\mathbf{X}}$ satisfies (iii) of Lemma 2.1, the rank of $\underline{\mathbf{X}}$ can be deduced from the Jordan normal form of $\mathbf{X}_2\mathbf{X}_1^{-1}$. This is stated in the following result, also due to Ja' Ja' [21].

LEMMA 2.2. *Let* $\underline{\mathbf{X}} \in \mathcal{R}_I$ *and suppose* $\mathbf{X}_2\mathbf{X}_1^{-1}$ *has* I *real eigenvalues. Let the Jordan normal form of* $\mathbf{X}_2\mathbf{X}_1^{-1}$ *be given by* $\operatorname{diag}(\lambda_1, \ldots, \lambda_p, \mathbf{J}_{m_1}(\mu_1), \ldots, \mathbf{J}_{m_r}(\mu_r))$, *where* $\mathbf{J}_{m_j}(\mu_j)$ *denotes an* $m_j \times m_j$ *Jordan block with diagonal elements equal to* μ_j *and* $m_j \geq 2$. *Then*

$$\operatorname{rank}_{\otimes}(\underline{\mathbf{X}}) = I + r. \tag{2.3}$$

For an eigenvalue λ_j of an $I \times I$ matrix $\mathbf{G}$, we define the *algebraic multiplicity* of λ_j as the multiplicity of λ_j as root of the characteristic polynomial $\det(\mathbf{G} - \lambda\mathbf{I}_I)$, and the *geometric multiplicity* of λ_j as the maximum number of linearly independent eigenvectors of $\mathbf{G}$ associated with λ_j (i.e., the dimensionality of the eigenspace of λ_j). Let $\mathbf{G} = \operatorname{diag}(\lambda_1, \ldots, \lambda_p, \mathbf{J}_{m_1}(\mu_1), \ldots, \mathbf{J}_{m_r}(\mu_r))$, with $m_j \geq 2$ for $j = 1, \ldots, r$. Then the eigenvalues of $\mathbf{G}$ are $\lambda_1, \ldots, \lambda_p, \mu_1, \ldots, \mu_r$ (not necessarily distinct), and each Jordan block $\mathbf{J}_{m_j}(\mu_j)$ adds m_j to the algebraic multiplicity of μ_j and 1 to the geometric multiplicity of μ_j. This establishes a relation between the eigenvalues of $\mathbf{X}_2\mathbf{X}_1^{-1}$ and the rank of the array $\underline{\mathbf{X}}$ in Lemma 2.2. In particular, if $\mathbf{X}_2\mathbf{X}_1^{-1}$ has I real eigenvalues and is diagonalizable, then $\underline{\mathbf{X}}$ has rank I, which is case (i) of Lemma 2.1.

When $I \times I \times 2$ arrays are randomly drawn from a continuous distribution, they have rank I or $I+1$, both with positive probability; see Ten Berge and Kiers [44]. Their *typical rank* is said to be $\{I, I+1\}$. A typical array $\underline{\mathbf{X}}$ of rank I satisfies (i) of Lemma 2.1, and $\mathbf{X}_2\mathbf{X}_1^{-1}$ has I distinct real eigenvalues. A typical array $\underline{\mathbf{X}}$ of rank $I+1$ satisfies (ii) of Lemma 2.1, and the eigenvalues of $\mathbf{X}_2\mathbf{X}_1^{-1}$ are again distinct.

If a three-way array of size $I \times J \times K$ has a one-valued typical rank, this is called its *generic rank*. In this case, a generic $I \times J \times K$ array has rank equal to its generic rank.

3. Diverging CP components for $I \times I \times 2$ arrays of rank $I+1$ and $R = I$. Here, we discuss the analysis of Stegeman [38] that shows how diverging CP components occur for typical $I \times I \times 2$ arrays of rank $I+1$ and $R = I$. Let

$$\mathcal{S}_I = \{\underline{\mathbf{Y}} \in \mathcal{R}_I : \underline{\mathbf{Y}} \text{ has rank at most } I\}. \tag{3.1}$$

Hence, the set $\mathcal{S}_I$ consists of the arrays in $\mathcal{R}_I$ which satisfy (i) of Lemma 2.1. Note that $\mathcal{S}_I$ contains only arrays of rank I, and not less than I, due to its restriction to $\mathcal{R}_I$.

Let $\underline{\mathbf{Z}} \in \mathcal{R}_I$ be typical and have rank $I+1$. Then $\underline{\mathbf{Z}}$ satisfies (ii) of Lemma 2.1. We consider the following CP problem:

$$\begin{aligned} &\text{Minimize} \|\underline{\mathbf{Z}} - \underline{\mathbf{Y}}\|^2 \\ &\text{subject to } \underline{\mathbf{Y}} \in \mathcal{S}_I. \end{aligned} \tag{3.2}$$

If problem (3.2) has an optimal solution $\underline{\mathbf{X}}$, then $\underline{\mathbf{X}}$ is a boundary point of $\mathcal{S}_I$. The following result defines the interior points and boundary points of $\mathcal{S}_I$ in $\mathcal{R}_I$ and is due to Stegeman [38].

LEMMA 3.1. *For $\underline{\mathbf{X}} \in \mathcal{R}_I$, the following statements hold:*

(a) *$\underline{\mathbf{X}}$ is an interior point of $\mathcal{S}_I$ if and only if $\mathbf{X}_2\mathbf{X}_1^{-1}$ has I distinct real eigenvalues.*

(b) *$\underline{\mathbf{X}}$ is a boundary point of $\mathcal{S}_I$ in $\mathcal{R}_I$ if and only if $\mathbf{X}_2\mathbf{X}_1^{-1}$ has I real eigenvalues but not all distinct.*

The boundary points in (b) can have rank I or rank $\geq I+1$, depending on whether $\mathbf{X}_2\mathbf{X}_1^{-1}$ is diagonalizable (type I) or not (type II); see Lemma 2.1. Hence, the set $\mathcal{S}_I$ is not a closed subset of $\mathcal{R}_I$, and the existence of an optimal solution for problem (3.2) is not guaranteed. If problem (3.2) has an optimal solution $\underline{\mathbf{X}}$, then it is a boundary point of type I.

Remark 3.2. For a typical $\underline{\mathbf{Z}}$ of rank $I+1$, problem (3.2) does not seem to have an optimal solution in practice. We conjecture the following explanation for this. For $m \geq 2$, define the sets of matrices

$$\begin{aligned} \mathcal{B}(\lambda_0, m) &= \{\mathbf{Y} \in \mathbb{R}^{I \times I} : \ \mathbf{Y} \text{ has eigenvalue } \lambda_0 \text{ with algebraic multiplicity } m\} \\ &= \mathcal{B}_1(\lambda_0, m) \cup \cdots \cup \mathcal{B}_m(\lambda_0, m), \end{aligned}$$

with

$$\mathcal{B}_l(\lambda_0, m) = \{\mathbf{Y} \in \mathcal{B}(\lambda_0, m) : \ \text{rank}(\mathbf{Y} - \lambda_0 \mathbf{I}_I) = I - l\}, \qquad l = 1, \ldots, m.$$

Due to the upper-semicontinuity of matrix rank, the set $\mathcal{B}_l(\lambda_0, m)$ lies dense in $\mathcal{B}_l(\lambda_0, m) \cup \cdots \cup \mathcal{B}_m(\lambda_0, m)$. For a boundary point $\underline{\mathbf{X}}$ of $\mathcal{S}_I$, all eigenvalues of $\mathbf{X}_2\mathbf{X}_1^{-1}$ are real and $\mathbf{X}_2\mathbf{X}_1^{-1} \in \mathcal{B}(\lambda_0, m)$ for some eigenvalue λ_0 and $m \geq 2$ (see Lemma 3.1 (b)). For a boundary point of type I (with rank I), it holds that $\mathbf{X}_2\mathbf{X}_1^{-1} \in \mathcal{B}_m(\lambda_0, m)$ for all multiple eigenvalues λ_0 of $\mathbf{X}_2\mathbf{X}_1^{-1}$. For a boundary point of type II (with rank at least $I+1$), it holds that $\mathbf{X}_2\mathbf{X}_1^{-1} \in \mathcal{B}_l(\lambda_0, m)$, with $l < m$ for some multiple eigenvalue λ_0 of $\mathbf{X}_2\mathbf{X}_1^{-1}$. From these observations, it follows that the set of boundary points of type II lies dense on the boundary of the set $\mathcal{S}_I$. As stated above, if problem (3.2) has an optimal solution, then it is a boundary point of type I. We conjecture that this implies that, for a typical array $\underline{\mathbf{Z}}$ of rank $I+1$, problem (3.2) has no optimal solution.

If problem (3.2) does not have an optimal solution, then the sequence of CP updates $\underline{\mathbf{Y}}^{(n)}$ converges to a boundary point $\underline{\mathbf{X}}$ of type II (i.e., with $\mathbf{X}_2\mathbf{X}_1^{-1}$ having I real eigenvalues and not diagonalizable) such that $\|\underline{\mathbf{Z}} - \underline{\mathbf{X}}\|^2$ equals the infimum of $\|\underline{\mathbf{Z}} - \underline{\mathbf{Y}}\|^2$ over $\mathcal{S}_I$. Stegeman [38] shows that when $\underline{\mathbf{Y}}^{(n)}$ converges to $\underline{\mathbf{X}}$, the sequence $\underline{\mathbf{Y}}^{(n)}$ features diverging components. This can be seen as follows. The boundary point $\underline{\mathbf{X}}$ satisfies (iii) of Lemma 2.1, and its rank, which is at least $I+1$, is given by Lemma 2.2. We assume $\underline{\mathbf{Y}}^{(n)}$ to be interior points of $\mathcal{S}_I$, i.e., $\mathbf{Y}_2^{(n)}(\mathbf{Y}_1^{(n)})^{-1}$ has I distinct real eigenvalues. Then $\underline{\mathbf{Y}}^{(n)}$ has a rank-I decomposition of the form (2.2). Moreover, for the k-ranks we have $k_{\mathbf{A}^{(n)}} = k_{\mathbf{B}^{(n)}} = I$ and $k_{\mathbf{C}^{(n)}} = 2$, and

Kruskal's condition (1.4) yields that the decomposition is essentially unique. By continuity, $\mathbf{Y}_2^{(n)}(\mathbf{Y}_1^{(n)})^{-1}$ converges to $\mathbf{X}_2\mathbf{X}_1^{-1}$. Denote the eigendecomposition of $\mathbf{Y}_2^{(n)}(\mathbf{Y}_1^{(n)})^{-1}$ by $\mathbf{K}^{(n)}\mathbf{\Lambda}^{(n)}(\mathbf{K}^{(n)})^{-1}$. The matrix $\mathbf{X}_2\mathbf{X}_1^{-1}$ has I real eigenvalues but is not diagonalizable, and we have $\mathbf{A}^{(n)} = \mathbf{K}^{(n)}$, $\mathbf{B}^{(n)} = ((\mathbf{K}^{(n)})^{-1}\mathbf{Y}_1^{(n)})^T$, $\mathbf{C}_1^{(n)} = \mathbf{I}_I$, and $\mathbf{C}_2^{(n)} = \mathbf{\Lambda}^{(n)}$. Let λ be an eigenvalue of $\mathbf{X}_2\mathbf{X}_1^{-1}$ with algebraic multiplicity strictly larger than its geometric multiplicity, and associated Jordan block of size $m \times m$. Then m columns of $\mathbf{A}^{(n)}$ converge to the same eigenvector (up to a sign change) of λ, the corresponding m columns of $\mathbf{B}^{(n)}$ tend to linear dependency and large magnitudes, and the m corresponding columns of $\mathbf{C}^{(n)}$ become nearly identical to $(1 \quad \lambda)^T$. The pattern of the m CP components is such that their sum does not blow up. Clearly, this is a case of diverging CP components as defined by (1.5)–(1.6).

The diverging CP components are related to the Jordan form of $\mathbf{X}_2\mathbf{X}_1^{-1}$ in the way described above. Hence, based on Lemma 2.2, one may conclude that the number of groups of diverging CP components equals the rank of the boundary array $\underline{\mathbf{X}}$ minus I.

To illustrate the phenomenon of diverging CP components as described above, we return to the example in (1.7). For this randomly sampled $4 \times 4 \times 2$ array $\underline{\mathbf{Z}}$, the matrix $\mathbf{Z}_2\mathbf{Z}_1^{-1}$ has one pair of complex eigenvalues. Hence, $\underline{\mathbf{Z}}$ is a typical $4 \times 4 \times 2$ array of rank 5. Trying to fit the CP model with $R = 4$, results in three diverging components, as shown in (1.8)–(1.10). Next, we compute the array $\underline{\mathbf{Y}}$ corresponding to the final CP update, i.e., $\mathbf{Y}_k = \mathbf{A}\,\mathbf{C}_k\,\mathbf{B}^T$ for $k = 1, 2$. This $\underline{\mathbf{Y}}$ is an approximation of the optimal boundary array $\underline{\mathbf{X}}$. For the eigenvalues of $\mathbf{Y}_2\mathbf{Y}_1^{-1}$, we get

$$-1.5431, \qquad 0.4395, \qquad 0.4925, \qquad 0.5427. \tag{3.3}$$

Hence, three eigenvalues are close together. This corresponds to the three diverging components in (1.8)–(1.10) as discussed above.

4. A simultaneous GSD. Here, we introduce the simultaneous GSD (SGSD) model for $I \times I \times K$ arrays and show that it always has an optimal solution. We also discuss a relation between the CP model and the SGSD model as presented in De Lathauwer, De Moor, and Vandewalle [6]. In matrix notation, the SGSD model for an array $\underline{\mathbf{Z}}$ is

$$\mathbf{Z}_k = \mathbf{Q}_a\,\mathbf{R}_k\,\mathbf{Q}_b^T + \mathbf{E}_k, \qquad k = 1, \ldots K, \tag{4.1}$$

where $\mathbf{Q}_a$ and $\mathbf{Q}_b$ are $I \times I$ orthonormal and $\mathbf{R}_k$ are $I \times I$ upper triangular $k = 1, \ldots, K$. The matrices $\mathbf{Q}_a$, $\mathbf{Q}_b$, and $\mathbf{R}_k$ are determined by minimizing the sum-of-squares of the residuals $\mathbf{E}_k$, $k = 1, \ldots, K$. For this purpose, a Jacobi-type algorithm is presented in [6], and Van der Veen and Paulraj [47] developed an extended QZ algorithm. Like the CP model, we consider the real-valued SGSD model.

Next, we show that the SGSD model, contrary to the CP model, always has an optimal solution. Our approach is analogous to Krijnen [23]. We make use of the following lemma, which can be found in Ortega and Rheinboldt [31, p. 104].

LEMMA 4.1. *Let $g : D \subset \mathbb{R}^q \to \mathbb{R}$, where D is unbounded. Then all level sets of g are bounded if and only if $g(\boldsymbol{\theta}_n) \to \infty$ whenever $\{\boldsymbol{\theta}_n\} \subset D$ and $\|\boldsymbol{\theta}_n\| \to \infty$.*

We define the parameter vector of the SGSD model as

$$\boldsymbol{\theta} = \mathrm{vec}(\mathrm{vec}(\mathbf{Q}_a), \mathrm{vec}(\mathbf{Q}_b), \mathrm{vec}(\mathbf{R}_1), \ldots, \mathrm{vec}(\mathbf{R}_K)).$$

Let $f(\boldsymbol{\theta})$ be the sum-of-squares of the residuals of the SGSD model. Since f is continuous, the level sets $L(\gamma) = \{\boldsymbol{\theta} : \ f(\boldsymbol{\theta}) \leq \gamma\}$ are closed. We have the following result.

PROPOSITION 4.2. *All level sets of f are bounded, and the SGSD model has an optimal solution.*

Proof. We have $\|\boldsymbol{\theta}\|^2 = 2I + \sum_{k=1}^{K} \|\mathbf{R}_k\|^2$. Hence, $\|\boldsymbol{\theta}_n\| \to \infty$ implies that $\|\mathbf{R}_k\| \to \infty$ for at least one k. Moreover,

$$f(\boldsymbol{\theta})^{1/2} = \sum_{k=1}^{K} \|\mathbf{Z}_k - \mathbf{Q}_a \mathbf{R}_k \mathbf{Q}_b^T\| \geq \sum_{k=1}^{K} \Big| \|\mathbf{Z}_k\| - \|\mathbf{Q}_a \mathbf{R}_k \mathbf{Q}_b^T\| \Big| = \sum_{k=1}^{K} \Big| \|\mathbf{Z}_k\| - \|\mathbf{R}_k\| \Big|,$$

which implies that $f(\boldsymbol{\theta}_n) \to \infty$ whenever $\|\boldsymbol{\theta}_n\| \to \infty$. From Lemma 4.1, it follows that all level sets of f are bounded. Since the level sets are also closed, f attains its infimum on any nonempty level set. This completes the proof. ∎

Next, we present a relation between the CP model and the SGSD model, which was partly proven by De Lathauwer, De Moor, and Vandewalle [6]. We have the following result.

LEMMA 4.3. *Let $\underline{\mathbf{X}} \in \mathbb{R}^{I \times I \times K}$. The following statements hold:*

(i) *If $\underline{\mathbf{X}}$ has a full CP decomposition with $R = I$, then $\underline{\mathbf{X}}$ has a full SGSD.*

(ii) *Suppose $\mathbf{X}_1$ is nonsingular. Then $\underline{\mathbf{X}}$ has a full CP decomposition with $R = I$ if and only if $\mathbf{X}_k \mathbf{X}_1^{-1}$, $k = 1, \ldots, K$ have a simultaneous eigendecomposition with only real eigenvalues. Moreover, the full CP decomposition of $\underline{\mathbf{X}}$ is essentially unique if and only if $k_{\mathbf{C}} \geq 2$.*

(iii) *Suppose $\mathbf{X}_1$ is nonsingular. If $\underline{\mathbf{X}}$ has an essentially unique full CP decomposition with $R = I$, then the indeterminacies in the full SGSD of $\underline{\mathbf{X}}$ are only due to the indeterminacies in the full CP decomposition of $\underline{\mathbf{X}}$.*

Proof. First, we show (i). We have $\mathbf{X}_k = \mathbf{A}\,\mathbf{C}_k\,\mathbf{B}^T$, $k = 1, \ldots, K$; see (1.3). Let $\mathbf{A} = \mathbf{Q}_a \mathbf{R}_a$ be a QR-decomposition of $\mathbf{A}$, with $\mathbf{Q}_a$ orthonormal and $\mathbf{R}_a$ upper triangular. Analogously, let $\mathbf{B} = \mathbf{Q}_b \mathbf{L}_b$ be a QL-decomposition of $\mathbf{B}$, with $\mathbf{Q}_b$ orthonormal and $\mathbf{L}_b$ lower triangular. Then $\mathbf{X}_k = \mathbf{Q}_a\,(\mathbf{R}_a \mathbf{C}_k \mathbf{L}_b^T)\,\mathbf{Q}_b^T$, $k = 1, \ldots, K$ is a full SGSD for $\underline{\mathbf{X}}$.

The first part of the proof of (ii) is due to De Lathauwer, De Moor, and Vandewalle [6]. Suppose $\underline{\mathbf{X}}$ has a full CP decomposition with $R = I$. Then we have $\mathbf{X}_k \mathbf{X}_1^{-1} = \mathbf{A}\,\mathbf{C}_k\,\mathbf{C}_1^{-1}\,\mathbf{A}^{-1}$, which is an eigendecomposition with real eigenvalues and shows that $\mathbf{X}_k \mathbf{X}_1^{-1}$, $k = 1, \ldots, K$ have a simultaneous eigendecomposition. Next, suppose $\mathbf{X}_k \mathbf{X}_1^{-1} = \mathbf{A}\,\mathbf{C}_k\,\mathbf{A}^{-1}$ for diagonal matrices $\mathbf{C}_k$, $k = 2, \ldots, K$. Then $\mathbf{X}_k = \mathbf{A}\,\mathbf{C}_k\,\mathbf{A}^{-1}\,\mathbf{X}_1$. Taking $\mathbf{C}_1 = \mathbf{I}_I$ and $\mathbf{B}^T = \mathbf{A}^{-1}\,\mathbf{X}_1$ now yields a full CP decomposition of $\underline{\mathbf{X}}$ with $R = I$.

In the CP decomposition of $\underline{\mathbf{X}}$, we have $k_{\mathbf{A}} = k_{\mathbf{B}} = I$. Hence, Kruskal's condition (1.4) for essential uniqueness is equivalent to $k_{\mathbf{C}} \geq 2$. See also Leurgans, Ross, and Abel [28]. Moreover, $k_{\mathbf{C}} \geq 2$ is also necessary for uniqueness as is shown in Stegeman and Sidiropoulos [42].

Next, we show (iii). From (ii), it follows that $\mathbf{X}_k \mathbf{X}_1^{-1} = \mathbf{A}\,\mathbf{C}_k\,\mathbf{A}^{-1}$, $k = 2, \ldots, K$, and $k_{\mathbf{C}} \geq 2$. From the full SGSD of $\underline{\mathbf{X}}$, we obtain that also $\mathbf{Q}_a^T \mathbf{X}_k \mathbf{X}_1^{-1} \mathbf{Q}_a = \mathbf{R}_k \mathbf{R}_1^{-1}$, $k = 2, \ldots, K$ have a simultaneous eigendecomposition $\mathbf{R}_a \mathbf{C}_k \mathbf{R}_a^{-1}$, with $\mathbf{R}_a$ upper triangular up to a column permutation. From Kruskal's condition (1.4), it follows that $\mathbf{R}_k = \mathbf{R}_a \mathbf{C}_k \mathbf{R}_b$, with $\mathbf{R}_b = \mathbf{R}_a^{-1} \mathbf{R}_1$ and $\mathbf{C}_1 = \mathbf{I}_I$, is an essentially unique full CP decomposition. Thus we have $\mathbf{X}_k \mathbf{X}_1^{-1} = \mathbf{Q}_a \mathbf{R}_a \mathbf{C}_k \mathbf{R}_a^{-1} \mathbf{Q}_a^T = \mathbf{A} \mathbf{C}_k \mathbf{A}^{-1}$, $k = 2, \ldots, K$, which implies $\mathbf{Q}_a \mathbf{R}_a = \mathbf{A}$ (since $k_{\mathbf{C}} \geq 2$). Looking at $\mathbf{X}_1^{-1} \mathbf{X}_k$, we get equivalently $\mathbf{Q}_b \mathbf{R}_b^T = \mathbf{B}$. Hence, there are no other indeterminacies in the full SGSD of $\underline{\mathbf{X}}$ than those implied by CP essential uniqueness. This completes the proof. ∎

From the proof of Lemma 4.3, it follows that a CP decomposition of $\underline{\mathbf{X}}$ (if it exists) can be obtained from its full SGSD by computing the simultaneous eigendecomposi-

tion of $\mathbf{R}_k\mathbf{R}_1^{-1}$, $k = 2, \ldots, K$. This method is analogous to the one proposed in De Lathauwer, De Moor, and Vandewalle [6]. For the case $I \leq K$ a different method is given in Van der Veen and Paulraj [47].

5. The GSD model for $I \times I \times 2$ arrays. Here, we consider the SGSD model for $I \times I \times 2$ arrays and discuss its relation with the CP model. Since a (complex-valued) SGSD for two slices ($K = 2$) is known as a GSD (see Golub and Van Loan [12]), we will use the abbreviation GSD. Next, we show which of the arrays in Lemma 2.1 have a full (real-valued) GSD.

LEMMA 5.1. *For $\underline{\mathbf{X}} \in \mathcal{R}_I$, the following statements hold:*

(i) *If $\mathbf{X}_2\mathbf{X}_1^{-1}$ has I real eigenvalues and is diagonalizable, then $\underline{\mathbf{X}}$ has a full GSD.*
(ii) *If $\mathbf{X}_2\mathbf{X}_1^{-1}$ has at least one pair of complex eigenvalues, then $\underline{\mathbf{X}}$ does not have a full GSD.*
(iii) *If $\mathbf{X}_2\mathbf{X}_1^{-1}$ has I real eigenvalues but is not diagonalizable, then $\underline{\mathbf{X}}$ has a full GSD.*

Proof. If (i) holds, then $\underline{\mathbf{X}}$ has a full CP decomposition with $R = I$ of the form (2.2). Hence, $\underline{\mathbf{X}}$ also has a full GSD. Next, suppose (ii) holds, and $\underline{\mathbf{X}}$ has a full GSD. Then $\mathbf{X}_2\mathbf{X}_1^{-1} = \mathbf{Q}_a\mathbf{R}_2\mathbf{R}_1^{-1}\mathbf{Q}_a^T$, and

$$\det(\mathbf{X}_2\mathbf{X}_1^{-1} - \lambda\,\mathbf{I}_I) = \det(\mathbf{Q}_a^T\mathbf{X}_2\mathbf{X}_1^{-1}\mathbf{Q}_a - \lambda\,\mathbf{I}_I) = \det(\mathbf{R}_2\mathbf{R}_1^{-1} - \lambda\,\mathbf{I}_I).$$

Since $\mathbf{R}_2\mathbf{R}_1^{-1}$ is upper triangular and has only real eigenvalues, it follows that also $\mathbf{X}_2\mathbf{X}_1^{-1}$ has only real eigenvalues. But this contradicts (ii). Therefore, $\underline{\mathbf{X}}$ has no full GSD if (ii) holds.

Next, suppose (iii) holds. Then $\mathbf{X}_2\mathbf{X}_1^{-1} = \mathbf{P}\,\mathbf{J}\,\mathbf{P}^{-1}$, where $\mathbf{J}$ is the Jordan normal form. Let $\mathbf{P} = \mathbf{Q}_a\mathbf{R}_a$ be a QR-decomposition of $\mathbf{P}$, and let $\mathbf{X}_1^T\mathbf{Q}_a = \mathbf{Q}_b\mathbf{L}_b$ be a QL-decomposition of $\mathbf{X}_1^T\mathbf{Q}_a$. Then

$$\text{(5.1)}\quad \mathbf{X}_2 = \mathbf{Q}_a\mathbf{R}_a\mathbf{J}\mathbf{R}_a^{-1}\mathbf{Q}_a^T\mathbf{X}_1 = \mathbf{Q}_a\,(\mathbf{R}_a\mathbf{J}\mathbf{R}_a^{-1}\mathbf{L}_b^T)\,\mathbf{Q}_b^T \quad \text{and} \quad \mathbf{X}_1 = \mathbf{Q}_a\,\mathbf{L}_b^T\,\mathbf{Q}_b^T$$

is a full GSD of $\underline{\mathbf{X}}$. This completes the proof. ∎

Note that a full GSD requires $\mathbf{R}_1$ and $\mathbf{R}_2$ to be upper triangular. This is not the same as the generalized real Schur decomposition (see Golub and Van Loan [12]), which always exists for two $I \times I$ matrices and which has $\mathbf{R}_1$ upper quasi-triangular.

As we see from Lemma 5.1, the arrays satisfying (iii) do not have a full CP decomposition with $R = I$ but do have a full GSD. Note that the CP decomposition of arrays satisfying (i) is essentially unique if and only if the eigenvalues of $\mathbf{X}_2\mathbf{X}_1^{-1}$ are distinct; see (ii) of Lemma 4.3.

Since (iii) of Lemma 4.3 does not apply to the GSD in (5.1), one may wonder what the uniqueness properties of (5.1) are. The Jordan form $\mathbf{J} = \operatorname{diag}(\lambda_1, \ldots, \lambda_p, \mathbf{J}_{m_1}(\mu_1), \ldots, \mathbf{J}_{m_r}(\mu_r))$ is unique up to the order of the Jordan blocks. If $\lambda_1, \ldots, \lambda_p, \mu_1, \ldots, \mu_r$ are distinct, then the columns of $\mathbf{P}$ are unique up to the same ordering and up to scaling. Suppose there is a second GSD, i.e., $\mathbf{X}_k = \widetilde{\mathbf{Q}}_a\,\widetilde{\mathbf{R}}_k\,\widetilde{\mathbf{Q}}_b^T$, $k = 1, 2$. Then there holds $\widetilde{\mathbf{R}}_2\,\widetilde{\mathbf{R}}_1^{-1} = (\widetilde{\mathbf{Q}}_a^T\mathbf{P})\,\mathbf{J}\,(\widetilde{\mathbf{Q}}_a^T\mathbf{P})^{-1}$. In fact, we have $\widetilde{\mathbf{Q}}_a^T\mathbf{P} = \widehat{\mathbf{R}}\,\mathbf{\Pi}$, with $\widehat{\mathbf{R}}$ upper triangular and $\mathbf{\Pi}$ a permutation. Then $\widetilde{\mathbf{R}}_2\,\widetilde{\mathbf{R}}_1^{-1} = \widehat{\mathbf{R}}\,(\mathbf{\Pi}\,\mathbf{J}\,\mathbf{\Pi}^T)\,\widehat{\mathbf{R}}$ is a Jordan form with a different ordering of the Jordan blocks. Hence, the GSD in (5.1) is unique up to the indeterminacies of the Jordan form of $\mathbf{X}_2\mathbf{X}_1^{-1}$.

6. Using the GSD model to avoid diverging CP components. Here, we show how the relation between the GSD and CP models for $I \times I \times 2$ arrays can be used to avoid the problems of diverging CP components discussed in section 3.

First, we establish a relation between the set of $I \times I \times 2$ arrays that have a full CP decomposition with $R = I$, i.e., the set $\mathcal{S}_I$ in (3.1), and the set of arrays that have a full GSD. Let

$$\mathcal{P}_I = \{\underline{\mathbf{Y}} \in \mathcal{R}_I : \underline{\mathbf{Y}} \text{ has a full GSD}\}. \tag{6.1}$$

Hence, the set $\mathcal{P}_I$ consists of the arrays satisfying either (i) or (iii) in Lemma 5.1. From Lemmas 2.1 and 5.1, it follows that $\mathcal{S}_I \subset \mathcal{P}_I$. Moreover, Lemmas 3.1 and 5.1 show that $\mathcal{P}_I$ is the closure of $\mathcal{S}_I$ in $\mathcal{R}_I$ and has the same interior points and boundary points as $\mathcal{S}_I$. For the boundary points $\underline{\mathbf{X}}$ of $\mathcal{P}_I$ and $\mathcal{S}_I$, the matrix $\mathbf{X}_2\mathbf{X}_1^{-1}$ has I real eigenvalues which are not all distinct; see Lemma 3.1. As explained in Remark 3.2, the boundary points $\underline{\mathbf{X}}$ of type II, i.e., with $\mathbf{X}_2\mathbf{X}_1^{-1}$ not diagonalizable, lie dense on the boundary of $\mathcal{P}_I$.

Let $\underline{\mathbf{Z}}$ be a typical $I \times I \times 2$ array of rank $I+1$, i.e., $\underline{\mathbf{Z}}$ satisfies (ii) of Lemma 5.1. Recall that the CP problem (3.2) for $\underline{\mathbf{Z}}$ usually does not have an optimal solution (see Remark 3.2). We define the analogue GSD problem as

$$\begin{aligned} &\text{Minimize} \|\underline{\mathbf{Z}} - \underline{\mathbf{Y}}\|^2 \\ &\text{subject to } \underline{\mathbf{Y}} \in \mathcal{P}_I. \end{aligned} \tag{6.2}$$

From the analysis in Stegeman [38], it follows that $\mathcal{P}_I$ is a closed subset of $\mathcal{R}_I$. Hence, the GSD problem (6.2) has an optimal solution, and a GSD algorithm finds an optimal solution $\underline{\mathbf{X}}$ of problem (6.2) in terms of its full GSD. We will assume that the optimal solution $\underline{\mathbf{X}}$ obtained for the GSD problem (6.2) is a boundary point of $\mathcal{P}_I$ of type II, i.e., $\mathbf{X}_2\mathbf{X}_1^{-1}$ has I real eigenvalues but is not diagonalizable. We conjecture (see Remark 3.2) that this is true almost everywhere for typical $\underline{\mathbf{Z}}$ of rank $I+1$.

From the observations above and our discussion in section 3, it follows that the optimal solution $\underline{\mathbf{X}}$ of the GSD problem (6.2), if it is unique, is the limit point of the sequence of CP updates (featuring diverging components) which attempts to converge to the (nonexisting) optimal solution of the CP problem (3.2).

Next, we show how to extract the nondiverging CP components from the optimal GSD solution. The limit point of the diverging CP components can be obtained from the optimal GSD solution as a Tucker3 part from the Jordan form of $\mathbf{X}_2\mathbf{X}_1^{-1}$ or as a smaller GSD part. These CP+Jordan and CP+GSD representations will be discussed in sections 6.1 and 6.3, respectively. In section 6.2, we show how the GSD solution can be decomposed into rank-1 terms using the CP+Jordan representation. Here, the number of rank-1 terms equals the rank of the solution array.

6.1. Optimal GSD solution in CP+Jordan form. Let $\underline{\mathbf{X}}$ be the optimal solution of the GSD problem (6.2). As explained above, we assume that $\mathbf{X}_2\mathbf{X}_1^{-1}$ has only real eigenvalues but is not diagonalizable. Next, we show how to obtain the nondiverging CP components from $\underline{\mathbf{X}}$ and write the limit points of the groups of diverging CP components in Jordan form. We have $\mathbf{X}_k = \mathbf{Q}_a\,\mathbf{R}_k\,\mathbf{Q}_b^T$, $k = 1, 2$ from a GSD algorithm. Since $\underline{\mathbf{X}} \in \mathcal{R}_I$, the matrices $\mathbf{R}_k$, $k = 1, 2$ are nonsingular. Let the Jordan normal form $\mathbf{P}\,\mathbf{J}\,\mathbf{P}^{-1}$ of $\mathbf{R}_2\mathbf{R}_1^{-1}$ be given by $\mathbf{J} = \text{diag}(\lambda_1, \ldots, \lambda_p, \mathbf{J}_{m_1}(\mu_1), \ldots, \mathbf{J}_{m_r}(\mu_r))$, where $\mathbf{J}_{m_j}(\mu_j)$ denotes an $m_j \times m_j$ Jordan block with $m_j \geq 2$, and $r \geq 1$. Note that the Jordan form $\mathbf{J}$ of $\mathbf{R}_2\mathbf{R}_1^{-1}$ is also the Jordan form of $\mathbf{X}_2\mathbf{X}_1^{-1}$. Hence, $\mathbf{R}_2\mathbf{R}_1^{-1}$ also has only real eigenvalues but is not diagonalizable.

Now the following decomposition of $\underline{\mathbf{X}}$ can be obtained. Let $\mathbf{C}_1 = \mathbf{I}_p$, $\mathbf{C}_2 = \text{diag}(\lambda_1, \ldots, \lambda_p)$, and let $\mathbf{A}$ contain the corresponding columns of $\mathbf{Q}_a\mathbf{P}$ and $\mathbf{B}^T$ the corresponding rows of $\mathbf{P}^{-1}\mathbf{R}_1\mathbf{Q}_b^T$. For the r Jordan blocks $\mathbf{J}_{m_j}$, let $\mathbf{K}_j$ contain the corresponding columns of $\mathbf{Q}_a\mathbf{P}$ and $\mathbf{L}_j^T$ the corresponding rows of $\mathbf{P}^{-1}\mathbf{R}_1\mathbf{Q}_b^T$. Then

$$\mathbf{X}_2 = \mathbf{A}\,\mathbf{C}_2\,\mathbf{B}^T + \sum_{j=1}^{r} \mathbf{K}_j\,\mathbf{J}_{m_j}\,\mathbf{L}_j^T, \tag{6.3}$$

$$\mathbf{X}_1 = \mathbf{A}\,\mathbf{C}_1\,\mathbf{B}^T + \sum_{j=1}^{r} \mathbf{K}_j\,\mathbf{I}_{m_j}\,\mathbf{L}_j^T. \tag{6.4}$$

Hence, we have decomposed the optimal GSD solution $\underline{\mathbf{X}}$ into a nondiverging CP part and r parts with a Jordan block $\mathbf{J}_{m_j}$ instead of a diagonal matrix. In this way, diverging CP components are avoided, i.e., the components $\mathbf{A}, \mathbf{K}_1, \ldots, \mathbf{K}_r$ are linearly independent (since they are columns of $\mathbf{Q}_a\mathbf{P}$), the components $\mathbf{B}, \mathbf{L}_1, \ldots, \mathbf{L}_r$ are linearly independent (since they are the rows of $\mathbf{P}^{-1}\mathbf{R}_1\mathbf{Q}_b^T$), and none of the elements in the decomposition tends to infinity. Note that each part of the decomposition (6.3)–(6.4) can be written in GSD form by using QR- and QL-decompositions as in the proof of (i) of Lemma 4.3.

If the eigenvalues $\lambda_1, \ldots, \lambda_p$ are distinct, then the CP-part of the representation (6.3)–(6.4) is essentially unique. Indeed, we have p components and k-ranks $k_{\mathbf{A}} = k_{\mathbf{B}} = p$ and $k_{\mathbf{C}} = 2$, and essential uniqueness follows from Kruskal's condition (1.4). From the uniqueness properties of the Jordan form of $\mathbf{R}_2\mathbf{R}_1^{-1}$ it follows that if $\mu_1, \ldots, \mu_r$ are distinct, then the representation of the non-CP part of (6.3)–(6.4) is unique up to the order of the Jordan blocks $\mathbf{J}_{m_j}$ and the scaling of the principal vectors in $\mathbf{P}$.

Although the decomposition (6.3)–(6.4) features not only rank-1 terms, it is essentially unique and may be interpretable to the researcher. From a computational as well as a practical point of view, this is a considerable improvement with respect to facing diverging CP components.

In practice, the matrix $\mathbf{R}_2\mathbf{R}_1^{-1}$ of the corresponding optimal GSD solution obtained from a GSD algorithm does not have exactly identical eigenvalues. To be able to "recognize" the identical eigenvalues of $\mathbf{R}_2\mathbf{R}_1^{-1}$ and their geometric multiplicities, the GSD algorithm must have a sufficiently small stopping criterion. The identical eigenvalues can then be estimated as the average of the ones which are "close together." The Jordan normal form of $\mathbf{R}_2\mathbf{R}_1^{-1}$ can be estimated by using, e.g., the method proposed in Golub and Wilkinson [11]. Below, we present the algorithm to obtain representation (6.3)–(6.4). The algorithm is formulated for general R (instead of $R = I$) in order to make it applicable to the $I \times J \times 2$ case as well (see section 7).

ALGORITHM FOR CP+JORDAN REPRESENTATION OF OPTIMAL GSD SOLUTION.

Input: Optimal GSD solution $\mathbf{X}_k = \mathbf{Q}_a\,\mathbf{R}_k\,\mathbf{Q}_b^T$, $k = 1, 2$, where $\mathbf{R}_2\mathbf{R}_1^{-1}$ has only real eigenvalues but is not diagonalizable.

Output: CP+Jordan representation (6.3)–(6.4).

1. Calculate the Jordan form $\mathbf{P}\,\mathbf{J}\,\mathbf{P}^{-1}$ of $\mathbf{R}_2\mathbf{R}_1^{-1}$, where $\mathbf{J} = \operatorname{diag}(\lambda_1, \ldots, \lambda_p, \mathbf{J}_{m_1}(\mu_1), \ldots, \mathbf{J}_{m_r}(\mu_r))$. Here, $\mathbf{J}_{m_j}(\mu_j)$ denotes an $m_j \times m_j$ Jordan block with $m_j \geq 2$, and $r \geq 1$.
2. Set $\mathbf{C}_1 = \mathbf{I}_p$, $\mathbf{C}_2 = \operatorname{diag}(\lambda_1, \ldots, \lambda_p)$. For eigenvalues $\lambda_1, \ldots, \lambda_p$, let $\mathbf{A}$ contain the corresponding columns of $\mathbf{Q}_a\mathbf{P}$ and $\mathbf{B}^T$ the corresponding rows of $\mathbf{P}^{-1}\mathbf{R}_1\mathbf{Q}_b^T$.
3. For Jordan block $\mathbf{J}_{m_j}$, let $\mathbf{K}_j$ contain the corresponding columns of $\mathbf{Q}_a\mathbf{P}$ and $\mathbf{L}_j^T$ the corresponding rows of $\mathbf{P}^{-1}\mathbf{R}_1\mathbf{Q}_b^T$, $j = 1, \ldots, r$.
4. The CP+Jordan representation (6.3)–(6.4) follows, with p nondiverging CP components in $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}_1$, $\mathbf{C}_2$ and r limit points of groups of diverging CP components (see section 3).

The following result states that (6.3)–(6.4) can be written as a Tucker3 model (1.2).

PROPOSITION 6.1. *Let the Jordan form of* $\mathbf{R}_2\mathbf{R}_1^{-1}$ *be given by* $\operatorname{diag}(\lambda_1,\ldots,\lambda_p, \mathbf{J}_{m_1}(\mu_1),\ldots,\mathbf{J}_{m_r}(\mu_r))$, *where* $\mathbf{J}_{m_j}(\mu_j)$ *denotes an* $m_j \times m_j$ *Jordan block with* $m_j \geq 2$. *Set* $M = p + r + 1$. *The decomposition* (6.3)–(6.4) *can be written as a Tucker*3 *model with an* $I \times I \times M$ *core array and component matrices*

$$[\mathbf{A} \,|\, \mathbf{K}_1 \,|\, \ldots \,|\, \mathbf{K}_r], \quad [\mathbf{B} \,|\, \mathbf{L}_1 \,|\, \ldots \,|\, \mathbf{L}_r], \quad \begin{bmatrix} 1 & \ldots & 1 & 0 & 1 & \ldots & 1 \\ \lambda_1 & \ldots & \lambda_p & 1 & \mu_1 & \ldots & \mu_r \end{bmatrix}. \tag{6.5}$$

The number of nonzeros in the core array equals $2I - (p + r)$.

Proof. The first p columns of the component matrices in (6.5) follow from the CP part in (6.3)–(6.4). Next, we consider a Jordan block $\mathbf{J}_m(\mu)$, with $m \geq 2$. The corresponding part in (6.3)–(6.4) can be written as

$$\sum_{i=1}^{m} \mathbf{k}_i \otimes \mathbf{l}_i \otimes \begin{pmatrix} 1 \\ \mu \end{pmatrix} + \sum_{i=1}^{m-1} \mathbf{k}_i \otimes \mathbf{l}_{i+1} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{6.6}$$

where $\mathbf{k}_i$ and $\mathbf{l}_i$ are the columns of the corresponding matrices $\mathbf{K}$ and $\mathbf{L}$, respectively. Hence, (6.6) uses the corresponding columns of the component matrices in (6.5) and adds $m + (m-1)$ nonzeros to the Tucker3 core array.

Since the CP part adds p nonzeros to the Tucker3 core array, the total number of nonzeros equals

$$p + \sum_{j=1}^{r}(2m_j - 1) = p - r + 2\sum_{j=1}^{r} m_j = p - r + 2(I - p) = 2I - (p + r). \tag{6.7}$$

This completes the proof. ∎

Note that the restricted Tucker3 model in Proposition 6.1 is unique up to the indeterminacies in the CP+Jordan representation (6.3)–(6.4).

The result of Proposition 6.1 is in line with Harshman [15], who explains diverging CP components for $2 \times 2 \times 2$ arrays as "Parafac trying to model Tucker variation." Paatero [33] also noticed that his constructed sequences of diverging CP components have a limit that can be written in Tucker3 form.

The decomposition (6.3)–(6.4) of $\underline{\mathbf{X}}$ into p rank-1 terms and r rank-$(m_j, m_j, 2)$ terms (i.e., the ranks of the vectors in the three modes are m_j, m_j, and 2) is an example of the *block-term decomposition* introduced in De Lathauwer [8].

Remark 6.2. Note that it is not our goal to find a CP+Tucker3 representation of $\underline{\mathbf{Z}}$, for which $\mathbf{Z}_2\mathbf{Z}_1^{-1}$ has some complex eigenvalues. Such a representation exists if the eigenvalues of $\mathbf{Z}_2\mathbf{Z}_1^{-1}$ are distinct and can be obtained from the transformation

$$\mathbf{Z}_2\mathbf{Z}_1^{-1} = \mathbf{K}\,\mathbf{\Lambda}\,\mathbf{K}^{-1}, \tag{6.8}$$

where $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1,\ldots,\lambda_s,\mathbf{\Gamma}_1,\ldots,\mathbf{\Gamma}_t)$ and $\mathbf{\Gamma}_i$ is 2×2 and corresponds to a pair of complex eigenvalues of $\mathbf{Z}_2\mathbf{Z}_1^{-1}$; see, e.g., Horn and Johnson [19]. Instead, it is our goal to find the limit point $\underline{\mathbf{X}}$ of the sequence of CP updates featuring diverging components, and (6.3)–(6.4) is a representation of that point $\underline{\mathbf{X}}$.

Next, we illustrate the CP+Jordan algorithm by revisiting the $4 \times 4 \times 2$ example in (1.7)–(1.10) that was also discussed at the end of section 3. Using the Jacobi algorithm of De Lathauwer, De Moor, and Vandewalle [6] with $R = 4$ and a convergence criterion

of `1e-9`, we obtain the following optimal GSD solution for $\underline{\mathbf{Z}}$ in (1.7):

$$\mathbf{Q}_a = \begin{bmatrix} 0.1279 & 0.8039 & -0.5519 & 0.1813 \\ -0.7946 & -0.1776 & -0.2749 & 0.5113 \\ -0.5895 & 0.3628 & 0.1606 & -0.7037 \\ 0.0690 & -0.4367 & -0.7708 & -0.4588 \end{bmatrix}, \tag{6.9}$$

$$\mathbf{Q}_b = \begin{bmatrix} -0.6328 & 0.3387 & -0.0964 & -0.6896 \\ 0.6382 & 0.5502 & -0.4778 & -0.2486 \\ 0.1774 & 0.5110 & 0.8406 & -0.0294 \\ -0.4011 & 0.5670 & -0.2363 & 0.6795 \end{bmatrix}, \tag{6.10}$$

$$\mathbf{R}_1 = \begin{bmatrix} -1.1875 & -1.3604 & 1.1724 & -1.5430 \\ 0 & -1.0758 & 0.4567 & -0.3733 \\ 0 & 0 & -0.9103 & -0.7889 \\ 0 & 0 & 0 & 1.5915 \end{bmatrix}, \tag{6.11}$$

$$\mathbf{R}_2 = \begin{bmatrix} 1.8323 & 0.0832 & 0.0009 & -0.0168 \\ 0 & -0.5285 & -2.5802 & -0.9022 \\ 0 & 0 & -0.4472 & 1.4516 \\ 0 & 0 & 0 & 0.7818 \end{bmatrix}. \tag{6.12}$$

The GSD algorithm terminated after 24 sweeps with an error sum-of-squares of 0.051016. The latter is less than the value of 0.051204 obtained by the CP algorithm in section 1, indicating that the GSD solution is closer to $\underline{\mathbf{Z}}$ than the final CP update. The sum-of-squares distance between the GSD solution $\mathbf{X}_k = \mathbf{Q}_a\,\mathbf{R}_k\,\mathbf{Q}_b^T$, $k = 1, 2$ and the final CP update $\mathbf{Y}_k = \mathbf{A}\,\mathbf{C}_k\,\mathbf{B}^T$, $k = 1, 2$ is only 3.2144e-7. For the GSD solution, the eigenvalues of $\mathbf{X}_2\mathbf{X}_1^{-1}$ are

$$-1.5430, \qquad 0.4912, \qquad 0.4912, \qquad 0.4912. \tag{6.13}$$

Hence, for the final CP update, the three eigenvalues of $\mathbf{Y}_2\mathbf{Y}_1$ that were close together in (3.3) have become identical in the limit point $\underline{\mathbf{X}}$.

Next, we apply the CP+Jordan algorithm to the obtained GSD solution above. For the CP-part, we obtain

$$\mathbf{A} = \begin{bmatrix} 0.1279 \\ -0.7946 \\ -0.5895 \\ 0.0690 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0.8259 \\ 0.0385 \\ -0.4005 \\ -0.3950 \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} 2.8918 \\ -4.4621 \end{bmatrix}, \tag{6.14}$$

where the columns of $\mathbf{A}$ and $\mathbf{B}$ are normalized to length 1. Comparing this to the final CP-update in (1.8)–(1.10), we see that (6.14) is the nondiverging CP component of the final CP update. For the non-CP part of the CP+Jordan representation, we obtain

$$\mathbf{K} = \begin{bmatrix} -0.6779 & -0.0194 & -0.0500 \\ 0.6690 & -0.0900 & -0.1191 \\ 0.1326 & -0.2662 & 0.1248 \\ 0.2744 & 0.3003 & 0.1064 \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} 0.8879 & -2.6832 & 5.3125 \\ 1.7433 & -2.6564 & 1.9149 \\ -0.7556 & 3.1066 & 0.2265 \\ 1.0387 & 1.3806 & -5.2348 \end{bmatrix}, \tag{6.15}$$

$$\mathbf{J} = \begin{bmatrix} 0.4912 & 1 & 0 \\ 0 & 0.4912 & 1 \\ 0 & 0 & 0.4912 \end{bmatrix}. \tag{6.16}$$

Hence, the limit point of the three diverging CP components is represented as (6.15)–(6.16).

6.2. A rank-revealing decomposition of the optimal GSD solution. Here we discuss how one may obtain a decomposition into rank-1 terms of the optimal GSD solution, where the number of rank-1 terms equals the rank of the solution array $\underline{\mathbf{X}}$. We make use of the CP+Jordan representation (6.3)–(6.4). The Tucker3 representation of $\underline{\mathbf{X}}$ in Proposition 6.1 decomposes $\underline{\mathbf{X}}$ into $2I-(p+r)$ rank-1 terms. Lemma 2.2 states that $\underline{\mathbf{X}}$ has rank $I+r$. Hence, the number of rank-1 terms in the Tucker3 representation is equal to $\text{rank}_{\otimes}(\underline{\mathbf{X}}) = I+r$ if and only if $I = p+2r$. That is, if all Jordan blocks $\mathbf{J}_{m_j}(\mu_j)$ have size $m_j = 2$. In this case, (6.3)–(6.4) itself is a rank-revealing decomposition of $\underline{\mathbf{X}}$. From (6.6), it follows that for $p = 0$, $r = 1$, and $m = 2$, the representation has the form (1.11) of De Silva and Lim [10] with $\mathbf{x}_1 = \mathbf{k}_1$, $\mathbf{x}_2 = \mathbf{l}_2$, $\mathbf{x}_3 = \binom{1}{\mu}$, $\mathbf{y}_1 = \mathbf{k}_2$, $\mathbf{y}_2 = \mathbf{l}_1$, and $\mathbf{y}_3 = \binom{0}{1}$. For general p and r and $m_j = 2$ for $j = 1, \ldots, r$, the representation (1.11) can be generalized to

$$\underline{\mathbf{X}} = \sum_{i=1}^{p} \mathbf{z}_1^{(i)} \otimes \mathbf{z}_2^{(i)} \otimes \mathbf{z}_3^{(i)} + \sum_{j=1}^{r} (\mathbf{x}_1^{(j)} \otimes \mathbf{x}_2^{(j)} \otimes \mathbf{y}_3 + \mathbf{x}_1^{(j)} \otimes \mathbf{y}_2^{(j)} \otimes \mathbf{x}_3^{(j)} + \mathbf{y}_1^{(j)} \otimes \mathbf{x}_2^{(j)} \otimes \mathbf{x}_3^{(j)}). \tag{6.17}$$

If there is a Jordan block $\mathbf{J}_{m_j}(\mu_j)$ with size $m_j \geq 3$, then the number of rank-1 terms in the decomposition (6.3)–(6.4) is larger than $\text{rank}_{\otimes}(\underline{\mathbf{X}}) = I+r$. However, a method to obtain a decomposition into $I+r$ rank-1 terms from (6.3)–(6.4) can be found in Ja' Ja' [21]. Consider the $m \times m \times 2$ array consisting of the slices $\mathbf{I}_m$ and $\mathbf{J}_m(\mu)$, with $m \geq 2$. From Lemma 2.2, it follows that this array has rank $m+1$. Let $\mathbf{w} = (w_0\ w_1\ \ldots w_{m-1})^T$, and let $\mathbf{e}_m$ denote the mth column of $\mathbf{I}_m$. Then $\mathbf{J}_m(\mu) - \mathbf{e}_m\,\mathbf{w}^T$ has characteristic polynomial $f(\lambda - \mu)$, with

$$f(x) = w_0 + w_1\,x + \cdots + w_{m-1}\,x^{m-1} - x^m. \tag{6.18}$$

It follows that we can pick $\mathbf{w}$ such that $f(x)$ has m distinct real roots. By Lemma 2.1, the $m \times m \times 2$ array with slices $\mathbf{I}_m$ and $\mathbf{J}_m(\mu) - \mathbf{e}_m\,\mathbf{w}^T$ has rank m, and a rank-m decomposition can be obtained from an eigendecomposition of its second slice. Since we have subtracted the rank-1 slice $\mathbf{e}_m\,\mathbf{w}^T$, this gives us a rank-$(m+1)$ decomposition of the array with $\mathbf{I}_m$ and $\mathbf{J}_m(\mu)$.

Applying this procedure to each Jordan block in (6.3)–(6.4) yields a decomposition of $\underline{\mathbf{X}}$ into $p + \sum_{j=1}^{r}(m_j+1) = I+r$ rank-1 terms. Since we have freedom in choosing the vector $\mathbf{w}$, this decomposition is not essentially unique (also for $m_j = 2$).

6.3. Optimal GSD solution in CP+GSD form. Since the Jordan form has a discontinuous transition from diagonalizable to nondiagonalizable matrices, it is numerically unstable, and the obtained Jordan form is extremely sensitive to tolerances for "recognizing" identical eigenvalues; see the discussion in Golub and Van Loan [12]. It follows from (6.3)–(6.4) that the complete non-CP part may also be represented in a full GSD of size $I-p$. For this, one only has to determine the eigenvalues of $\mathbf{X}_2\mathbf{X}_1^{-1}$ with algebraic multiplicity equal to 1, which is numerically more stable. Here, we show how the CP part and the GSD of the non-CP part can be computed without first computing the Jordan representation (6.3)–(6.4). Also, if one is only interested in obtaining the nondiverging CP components, computing the CP-part of the CP+GSD representation is an efficient way.

We assume that the optimal GSD solution $\underline{\mathbf{X}}$ has been obtained from a GSD algorithm and the GSD $\mathbf{X}_k = \mathbf{Q}_a \mathbf{R}_k \mathbf{Q}_b^T$, $k = 1, 2$ is known. Let the Jordan form of $\mathbf{R}_2\mathbf{R}_1^{-1}$ be given by $\mathbf{J} = \text{diag}(\lambda_1, \ldots, \lambda_p, \mathbf{J}_{m_1}(\mu_1), \ldots, \mathbf{J}_{m_r}(\mu_r))$, where $\mathbf{J}_{m_j}(\mu_j)$ denotes an $m_j \times m_j$ Jordan block with $m_j \geq 2$, and $r \geq 1$. The eigenvalues and the Jordan form $\mathbf{J}$ of $\mathbf{R}_2\mathbf{R}_1^{-1}$ are the same as those of $\mathbf{X}_2\mathbf{X}_1^{-1}$ and those of $\mathbf{R}_1^{-1}\mathbf{R}_2$. As mentioned above, we assume that $\mathbf{R}_2\mathbf{R}_1^{-1}$ has only real eigenvalues but is not diagonalizable. In the following, we assume that the eigenvalues $\lambda_1, \ldots, \lambda_p$ are known. For this, it is not necessary to compute the complete Jordan form $\mathbf{J}$.

First, we show how to obtain the CP-part of (6.3)–(6.4). For simplicity, we assume that none of the eigenvalues μ_j is equal to a λ_i. Let $\mathbf{R}_a$ have as columns the eigenvectors of $\mathbf{R}_2\mathbf{R}_1^{-1}$ corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_p$. From the discussion in section 6.1, it follows that $\mathbf{A} = \mathbf{Q}_a \mathbf{R}_a$. Next, we find $\mathbf{B}$. If $\mathbf{R}_2\mathbf{R}_1^{-1} = \mathbf{P}\,\mathbf{J}\,\mathbf{P}^{-1}$, then $\mathbf{R}_1^{-1}\mathbf{R}_2 = (\mathbf{R}_1^{-1}\mathbf{P})\,\mathbf{J}\,(\mathbf{R}_1^{-1}\mathbf{P})^{-1}$. Let $\mathbf{R}_b$ have as rows the left eigenvectors of $\mathbf{R}_1^{-1}\mathbf{R}_2$ corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_p$, i.e., $\mathbf{R}_b\,\mathbf{R}_1^{-1}\mathbf{R}_2 = \text{diag}(\lambda_1, \ldots, \lambda_p)\,\mathbf{R}_b$. Then we have $\mathbf{B} = \mathbf{Q}_b\,\mathbf{R}_b^T$ (see section 6.1). Since $\mathbf{R}_2\mathbf{R}_1^{-1}$ and $\mathbf{R}_1^{-1}\mathbf{R}_2$ are upper triangular, the columns of $\mathbf{R}_a$ and the rows of $\mathbf{R}_b$ are the columns and rows, respectively, of an $I \times I$ upper triangular matrix. We normalize the rows of $\mathbf{R}_b$ such that the first nonzero element becomes 1, and we normalize the columns of $\mathbf{R}_a$ such that the last nonzero element becomes 1. Let $\mathbf{C}_k$ be the $p \times p$ diagonal matrix containing the diagonal elements of $\mathbf{R}_k$ corresponding to the locations of $\lambda_1, \ldots, \lambda_p$ on the diagonal of $\mathbf{R}_2\mathbf{R}_1^{-1}$. It now follows that the CP-part in (6.3)–(6.4) is equal (up to scaling/rescaling and a joint permutation of the p CP components) to $\mathbf{A}\,\mathbf{C}_k\,\mathbf{B}^T$, $k = 1, 2$.

Note that the eigenvalues $\lambda_1, \ldots, \lambda_p$ may appear anywhere on the diagonal of $\mathbf{R}_2\mathbf{R}_1^{-1}$. Hence, unlike the ordering in the Jordan form $\mathbf{J}$, the eigenvalues λ_j do not need to appear as the first p diagonal elements of $\mathbf{R}_2\mathbf{R}_1^{-1}$. This is due to the permutation indeterminacy of the GSD solution. See also the discussion at the end of section 5.

Next, we show how to obtain the GSD of the non-CP part of (6.3)–(6.4). Define $\mathbf{T}_k = \mathbf{R}_k - \mathbf{R}_a\,\mathbf{C}_k\,\mathbf{R}_b$, $k = 1, 2$. Then $\mathbf{Y}_k = \mathbf{Q}_a\,\mathbf{T}_k\,\mathbf{Q}_b^T$, $k = 1, 2$ is the non-CP part of (6.3)–(6.4). From (6.3)–(6.4), it follows that $\mathbf{Y}_1 = \mathbf{K}\,\mathbf{I}_{I-p}\,\mathbf{L}^T$ and $\mathbf{Y}_2 = \mathbf{K}\,\tilde{\mathbf{J}}\,\mathbf{L}^T$ for $I \times (I-p)$ matrices $\mathbf{K}$ and $\mathbf{L}$ of full column rank and an $(I-p) \times (I-p)$ Jordan form $\tilde{\mathbf{J}}$. This implies that $\mathbf{Y}_1$ and $\mathbf{Y}_2$ have rank $I-p$ and identical column and row spaces. These properties of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ also hold for $\mathbf{T}_1$ and $\mathbf{T}_2$. Moreover, by definition, $\mathbf{T}_k$ is upper triangular and has zeros on the diagonal corresponding to the locations of $\lambda_1, \ldots, \lambda_p$ on the diagonal of $\mathbf{R}_2\mathbf{R}_1^{-1}$. From the I locations on the diagonal of $\mathbf{R}_2\mathbf{R}_1^{-1}$, let $1 \leq i_1 < i_2 < \cdots < i_{I-p} \leq I$ be those not containing $\lambda_1, \ldots, \lambda_p$. Let $\widetilde{\mathbf{T}}_k$ contain the columns $i_1, i_2, \ldots, i_{I-p}$ of $\mathbf{T}_k$, in the same order as they appear in $\mathbf{T}_k$, $k = 1, 2$. Then each of these columns has a nonzero diagonal element in $\mathbf{T}_k$, and since $\mathbf{T}_k$ is upper triangular, $\widetilde{\mathbf{T}}_k$ has rank $I-p$. Since $\mathbf{T}_k$ also has rank $I-p$, it follows that the column spaces of $\widetilde{\mathbf{T}}_k$ and $\mathbf{T}_k$ are identical. Also, the column spaces of $\widetilde{\mathbf{T}}_1$ and $\widetilde{\mathbf{T}}_2$ are identical. We write $\mathbf{T}_k = \widetilde{\mathbf{T}}_k\,\mathbf{H}_k^T$, where $\mathbf{H}_k = \mathbf{T}_k^T\,\widetilde{\mathbf{T}}_k\,(\widetilde{\mathbf{T}}_k^T\widetilde{\mathbf{T}}_k)^{-1}$, $k = 1, 2$. We need the following lemmas.

LEMMA 6.3. *There holds* $\widetilde{\mathbf{T}}_k = \widetilde{\mathbf{Q}}\,\widetilde{\mathbf{R}}_k$ *for some* $I \times (I-p)$ *columnwise orthonormal* $\widetilde{\mathbf{Q}}$ *and some* $(I-p) \times (I-p)$ *upper triangular* $\widetilde{\mathbf{R}}_k$, $k = 1, 2$.

Proof. Let $\widetilde{\mathbf{T}}_k$ contain columns $i_1, \ldots, i_{I-p}$ of $\mathbf{T}_k$, with $1 \leq i_1 < i_2 < \cdots < i_{I-p} \leq I$. Let $\mathbf{t}_{i_n}^{(k)}$ denote column i_n of $\mathbf{T}_k$, which is column n of $\widetilde{\mathbf{T}}_k$, $k = 1, 2$. Then $\mathbf{t}_{i_n}^{(k)}$ has the last $I - i_n$ elements equal to zero and element i_n nonzero. We obtain $\widetilde{\mathbf{Q}}$ and $\widetilde{\mathbf{R}}_1$ from a QR-decomposition of $\widetilde{\mathbf{T}}_1$ by means of the Gram–Schmidt process.

Let

$$\mathbf{v}_1 = \mathbf{t}_{i_1}^{(1)} \qquad \text{and} \qquad \mathbf{v}_n = \mathbf{t}_{i_n}^{(1)} - \sum_{j=1}^{n-1} \text{proj}_{\mathbf{v}_j}\, \mathbf{t}_{i_n}^{(1)}, \qquad n = 2, \ldots, I-p, \tag{6.19}$$

where $\text{proj}_{\mathbf{v}}\, \mathbf{t} = (\mathbf{t}^T\mathbf{v})/(\mathbf{v}^T\mathbf{v})\, \mathbf{v}$ denotes the orthogonal projection of $\mathbf{t}$ onto $\mathbf{v}$. The columns of $\widetilde{\mathbf{Q}}$ are the unit length versions of $\mathbf{v}_1, \ldots, \mathbf{v}_{I-p}$, and the elements of $\widetilde{\mathbf{R}}_1$ follow from (6.19).

The columns of $\widetilde{\mathbf{Q}}$ form an orthonormal basis for the column space of $\widetilde{\mathbf{T}}_1$ and, hence, also for the column space of $\widetilde{\mathbf{T}}_2$. This implies that for every column $\mathbf{t}_{i_n}^{(2)}$, there is a vector $\mathbf{w}$ such that

$$\mathbf{t}_{i_n}^{(2)} = \widetilde{\mathbf{Q}}\, \mathbf{w}. \tag{6.20}$$

From (6.19), it follows that column j of $\widetilde{\mathbf{Q}}$ has the last $I - i_j$ elements equal to zero and element i_j nonzero, $j = 1, \ldots, I-p$. Since $\mathbf{t}_{i_n}^{(2)}$ has the last $I - i_n$ elements equal to zero, (6.20) implies that $\mathbf{t}_{i_n}^{(2)}$ lies in the space spanned by the first n columns of $\widetilde{\mathbf{Q}}$. Hence, $\mathbf{w}$ in (6.20) has the last $I - p - n$ elements equal to zero. It follows that $\widetilde{\mathbf{T}}_2 = \widetilde{\mathbf{Q}}\, \widetilde{\mathbf{R}}_2$ for some $(I-p) \times (I-p)$ upper triangular $\widetilde{\mathbf{R}}_2$. Since $\widetilde{\mathbf{Q}}$ has full column rank, the matrix $\widetilde{\mathbf{R}}_2$ is uniquely determined. This completes the proof. ∎

Recall that, since the column spaces of $\widetilde{\mathbf{T}}_k$ and $\mathbf{T}_k$ are identical, we may write $\mathbf{T}_k = \widetilde{\mathbf{T}}_k\, \mathbf{H}_k^T$, $k = 1, 2$.

LEMMA 6.4. *Let* $\mathbf{H}_k$ *satisfy* $\mathbf{T}_k = \widetilde{\mathbf{T}}_k\, \mathbf{H}_k^T$, $k = 1, 2$. *Then* $\mathbf{H}_1 = \mathbf{H}_2$.

Proof. Since $\widetilde{\mathbf{T}}_k$ contains columns $i_1, \ldots, i_{I-p}$ of $\mathbf{T}_k$, it follows that rows $i_1, \ldots, i_{I-p}$ of $\mathbf{H}_k$ are equal to rows $1, \ldots, I-p$ of $\mathbf{I}_{I-p}$, $k = 1, 2$. Let the row permutation $\mathbf{\Pi}$ be such that $\mathbf{\Pi}\, \mathbf{H}_k = \begin{bmatrix} \mathbf{I}_{I-p} \\ \widetilde{\mathbf{H}}_k \end{bmatrix}$.

Since $\mathbf{H}_k$ has full column rank, $\text{rank}(\mathbf{T}_k) = I - p$, and $\mathbf{T}_k^T = \mathbf{H}_k\, \widetilde{\mathbf{T}}_k^T$, the column space of $\mathbf{H}_k$ is identical to the column space of $\mathbf{T}_k^T$. Moreover, since the row spaces of $\mathbf{T}_1$ and $\mathbf{T}_2$ are identical, the column spaces of $\mathbf{H}_1$ and $\mathbf{H}_2$ are also identical. It follows that each column of $\mathbf{\Pi}\, \mathbf{H}_1$ must lie in the column space of $\mathbf{\Pi}\, \mathbf{H}_2$. Since both matrices have $\mathbf{I}_{I-p}$ as their first $I-p$ rows, this yields that $\mathbf{\Pi}\, \mathbf{H}_1 = \mathbf{\Pi}\, \mathbf{H}_2$. Hence, $\mathbf{H}_1 = \mathbf{H}_2$, which completes the proof. ∎

Using Lemmas 6.3 and 6.4, the GSD of the non-CP part of (6.3)–(6.4) can be computed as follows. From a QR-decomposition of $\widetilde{\mathbf{T}}_1$, we obtain $\widetilde{\mathbf{T}}_1 = \widetilde{\mathbf{Q}}\, \widetilde{\mathbf{R}}_1$. The matrix $\widetilde{\mathbf{R}}_2$ in Lemma 6.3 follows from $\widetilde{\mathbf{R}}_2 = \widetilde{\mathbf{Q}}^T \widetilde{\mathbf{T}}_2$. The matrix $\mathbf{H} = \mathbf{H}_1 = \mathbf{H}_2$ in Lemma 6.4 is obtained as $\mathbf{H} = \mathbf{T}_1^T\, \widetilde{\mathbf{T}}_1\, (\widetilde{\mathbf{T}}_1^T \widetilde{\mathbf{T}}_1)^{-1}$. Next, let $\mathbf{H} = \widehat{\mathbf{Q}}\, \widehat{\mathbf{R}}^T$ be a QL-decomposition of $\mathbf{H}$ with an $I \times (I-p)$ columnwise orthonormal $\widehat{\mathbf{Q}}$ and an $(I-p) \times (I-p)$ upper triangular $\widehat{\mathbf{R}}$. It follows that

$$\mathbf{Y}_k = \mathbf{Q}_a\, \mathbf{T}_k\, \mathbf{Q}_b^T = \mathbf{Q}_a\, \widetilde{\mathbf{T}}_k\, \mathbf{H}^T\, \mathbf{Q}_b^T = (\mathbf{Q}_a\, \widetilde{\mathbf{Q}})\, (\widetilde{\mathbf{R}}_k\, \widehat{\mathbf{R}})\, (\mathbf{Q}_b\, \widehat{\mathbf{Q}})^T, \qquad k = 1, 2 \tag{6.21}$$

is a full GSD of size $I-p$ of the non-CP part of (6.3)–(6.4). Below, we present the algorithm to obtain a CP+GSD representation of the optimal GSD solution $\underline{\mathbf{X}}$. The algorithm is formulated for general R (instead of $R = I$) in order to make it applicable to the $I \times J \times 2$ case as well (see section 7).

ALGORITHM FOR CP+GSD REPRESENTATION OF OPTIMAL GSD SOLUTION.

Input: Optimal GSD solution $\mathbf{X}_k = \mathbf{Q}_a\,\mathbf{R}_k\,\mathbf{Q}_b^T$, $k = 1, 2$, where $\mathbf{R}_2\mathbf{R}_1^{-1}$ has only real eigenvalues but is not diagonalizable.

Output: CP+GSD representation $\mathbf{X}_k = \mathbf{A}\,\mathbf{C}_k\,\mathbf{B}^T + \mathbf{Q}_1\,\mathbf{R}_k^{(0)}\,\mathbf{Q}_2^T$, $k = 1, 2$.

1. Calculate the eigenvalues $\lambda_1, \ldots, \lambda_p$ of $\mathbf{R}_2\mathbf{R}_1^{-1}$ with algebraic multiplicity 1. Set $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$.
2. Determine $\mathbf{R}_a$ $(R \times p)$ as $\mathbf{R}_2\mathbf{R}_1^{-1}\,\mathbf{R}_a = \mathbf{R}_a\,\mathbf{\Lambda}$. Normalize the columns of $\mathbf{R}_a$ such that the last nonzero element becomes 1. Determine $\mathbf{R}_b$ $(p \times R)$ as $\mathbf{R}_b\,\mathbf{R}_1^{-1}\mathbf{R}_2 = \mathbf{\Lambda}\,\mathbf{R}_b$. Normalize the rows of $\mathbf{R}_b$ such that the first nonzero element becomes 1.
3. Set $\mathbf{A} = \mathbf{Q}_a\mathbf{R}_a$ and $\mathbf{B} = \mathbf{Q}_b\mathbf{R}_b^T$. Let $\mathbf{C}_k$ be the diagonal matrix containing the diagonal elements of $\mathbf{R}_k$ corresponding to the locations of $\lambda_1, \ldots, \lambda_p$ on the diagonal of $\mathbf{R}_2\mathbf{R}_1^{-1}$, $k = 1, 2$. The p nondiverging CP components are now obtained as $\mathbf{A}\,\mathbf{C}_k\,\mathbf{B}^T$, $k = 1, 2$.
4. Set $\mathbf{T}_k = \mathbf{R}_k - \mathbf{R}_a\,\mathbf{C}_k\,\mathbf{R}_b$, $k = 1, 2$. Let $\widetilde{\mathbf{T}}_k$ contain the $R - p$ columns of $\mathbf{T}_k$ with a nonzero diagonal element, in the same order as they appear in $\mathbf{T}_k$, $k = 1, 2$.
5. Compute the QR-decomposition $\widetilde{\mathbf{T}}_1 = \widetilde{\mathbf{Q}}\,\widetilde{\mathbf{R}}_1$ and set $\widetilde{\mathbf{R}}_2 = \widetilde{\mathbf{Q}}^T\widetilde{\mathbf{T}}_2$.
6. Set $\mathbf{H} = \mathbf{T}_1^T\,\widetilde{\mathbf{T}}_1\,(\widetilde{\mathbf{T}}_1^T\widetilde{\mathbf{T}}_1)^{-1}$ and compute the QL-decomposition $\mathbf{H} = \widehat{\mathbf{Q}}\,\widehat{\mathbf{R}}^T$.
7. Set $\mathbf{Q}_1 = \mathbf{Q}_a\,\widetilde{\mathbf{Q}}$, $\mathbf{Q}_2 = \mathbf{Q}_b\,\widehat{\mathbf{Q}}$ and $\mathbf{R}_k^{(0)} = \widetilde{\mathbf{R}}_k\,\widehat{\mathbf{R}}$, $k = 1, 2$. The size-$(R - p)$ GSD representation of the limit point of the diverging CP components is now obtained as $\mathbf{Q}_1\,\mathbf{R}_k^{(0)}\,\mathbf{Q}_2^T$, $k = 1, 2$.

To illustrate the CP+GSD algorithm, we return once again to the $4\times4\times2$ example in (1.7)–(1.10) that was also discussed at the end of sections 3 and 6.1. We apply the CP+GSD algorithm to the optimal GSD solution (6.9)–(6.12) for $\underline{\mathbf{Z}}$ in (1.7). For the CP-part, we obtain

$$\mathbf{R}_a = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{R}_b^T = \begin{bmatrix} 1.0000 \\ 0.3111 \\ 0.8314 \\ 2.0353 \end{bmatrix}, \quad \mathbf{C}_1 = -1.1875, \quad \mathbf{C}_2 = 1.8323. \tag{6.22}$$

Up to scaling/rescaling, the CP-part $\mathbf{A} = \mathbf{Q}_a\mathbf{R}_a$, $\mathbf{B} = \mathbf{Q}_b\mathbf{R}_b^T$, $\mathbf{C}_1$, $\mathbf{C}_2$ in (6.22) is equal to the CP-part (6.14) of the CP+Jordan representation. For the GSD-part, we obtain

$$\mathbf{Q}_1 = \begin{bmatrix} -0.6779 & -0.0470 & 0.2357 \\ 0.6690 & -0.2187 & 0.5479 \\ 0.1326 & -0.6466 & -0.6812 \\ 0.2744 & 0.7293 & -0.4244 \end{bmatrix}, \quad \mathbf{Q}_2 = \begin{bmatrix} 0.3387 & -0.0964 & -0.6896 \\ 0.5502 & -0.4778 & -0.2486 \\ 0.5110 & 0.8406 & -0.0294 \\ 0.5670 & -0.2363 & 0.6795 \end{bmatrix}, \tag{6.23}$$

$$\mathbf{R}_1^{(0)} = \begin{bmatrix} 1.4627 & -1.7991 & -0.3176 \\ 0 & 1.5699 & 1.1872 \\ 0 & 0 & 1.5963 \end{bmatrix}, \quad \mathbf{R}_2^{(0)} = \begin{bmatrix} 0.7185 & 2.9293 & 3.2016 \\ 0 & 0.7712 & -2.5885 \\ 0 & 0 & 0.7841 \end{bmatrix}. \tag{6.24}$$

Hence, the limit point of the three diverging CP components is represented as (6.23)–(6.24).

TABLE 7.1
Conjectures of Stegeman [40] on the occurrence of diverging CP components for a generic $I \times J \times 2$ array $\underline{\mathbf{Z}}$. Here, $I \geq J \geq 2$ and $R \geq 2$.

Case	$\underline{\mathbf{Z}} \in \mathbb{R}^{I \times J \times 2}$	$\text{Rank}_{\otimes}(\underline{\mathbf{Z}})$	R	Diverging CP components
2	$I = J$	$I+1$	$R = I$	almost everywhere
3	$I = J$	$I+1$	$R < I$	positive volume
5	$I = J$	I	$R < I$	positive volume
8	$I > J$	$\min(I, 2J)$	$R = J$	positive volume
9	$I > J$	$\min(I, 2J)$	$R < J$	positive volume

7. Extension to $I \times J \times 2$ arrays and general R. Stegeman [40] has mathematically analyzed the cases of diverging CP components occurring for generic $I \times J \times 2$ arrays and all values of R. The cases in which diverging components occur are listed in Table 7.1, as well as the conjectures of Stegeman [40] on the frequency of their occurrence. A generic $I \times J \times 2$ array $\underline{\mathbf{Z}}$ has rank $\min(I, 2J)$ if $I > J$, and rank I or $I+1$ (both on a set of positive volume) if $I = J$; see Ten Berge and Kiers [44].

It is shown in [40] that the cases of diverging CP components in Table 7.1 can be transformed to Case 2 (with $I = J = R$), which we have considered so far. Next, we extend our results in the previous sections by showing that in all cases in Table 7.1, the GSD approach may be used to avoid the problem of diverging CP components. Analogous to our previous results, the optimal GSD solution is the limit point of the sequence of CP updates (whether it features diverging components or not) and may be decomposed into a nondiverging CP-part and a Jordan part or into a nondiverging CP-part and a smaller GSD part.

The GSD model for an $I \times J \times 2$ array $\underline{\mathbf{Z}}$ is

$$\mathbf{Z}_k = \mathbf{Q}_a \, \mathbf{R}_k \, \mathbf{Q}_b^T + \mathbf{E}_k, \qquad k = 1, 2, \tag{7.1}$$

where $\mathbf{Q}_a$ ($I \times R$) and $\mathbf{Q}_b$ ($J \times R$) are columnwise orthonormal and $\mathbf{R}_k$ are $R \times R$ upper triangular, $k = 1, 2$. Without loss of generality, we assume $I \geq J$. Also, we assume $R \leq J$ (and $R \leq I$) and $R < \text{rank}_{\otimes}(\underline{\mathbf{Z}})$. From Table 7.1, it can be seen that this includes all cases. Finding $\mathbf{Q}_a$, $\mathbf{Q}_b$, $\mathbf{R}_1$, and $\mathbf{R}_2$, which minimize the sum-of-squares of the residuals in (7.1), can be achieved by a modification of the Jacobi algorithm of De Lathauwer, De Moor, and Vandewall [6]. This will be explained in section 7.1 below.

The proof of Proposition 4.2 can be used to show that the GSD model (7.1) for $I \times J \times 2$ arrays always has an optimal solution. Analogous to (6.1)–(6.2), we define

$$\mathcal{P}_{(I,J,R)} = \{\underline{\mathbf{Y}} \in \mathbb{R}^{I \times J \times 2} : \underline{\mathbf{Y}} \text{ has a full GSD (7.1) with } \mathbf{R}_1 \text{ and } \mathbf{R}_2 \text{ nonsingular}\} \tag{7.2}$$

and the GSD problem

$$\begin{aligned} &\text{Minimize} \quad \|\underline{\mathbf{Z}} - \underline{\mathbf{Y}}\|^2 \\ &\text{subject to} \underline{\mathbf{Y}} \in \mathcal{P}_{(I,J,R)}. \end{aligned} \tag{7.3}$$

From the analysis in Stegeman [40], it follows that the set $\mathcal{P}_{(I,J,R)}$ is closed, and hence, problem (7.3) always has an optimal solution. In Cases 3, 5, 8, and 9 of Table 7.1, the boundary of $\mathcal{P}_{(I,J,R)}$ is the set $\mathcal{P}_{(I,J,R)}$ itself, and the optimal solution $\underline{\mathbf{X}}$ of the GSD problem (7.3) has $\mathbf{R}_2 \mathbf{R}_1^{-1}$ with only real eigenvalues, some of which are identical. The problem of diverging CP components occurs if $\mathbf{R}_2 \mathbf{R}_1^{-1}$ is not diagonalizable; see [40]. In this case, the GSD of $\underline{\mathbf{X}}$ cannot be fully transformed to a CP representation, and

sequences of CP updates converging to $\underline{\mathbf{X}}$ feature diverging components. As in Case 2 of Table 7.1, two alternatives are a CP+Jordan or a CP+GSD representation. These can be obtained by using the algorithms in sections 6.1 and 6.3, respectively. As explained in section 6.2, a decomposition of $\underline{\mathbf{X}}$ into $\text{rank}_{\otimes}(\underline{\mathbf{X}})$ terms of rank 1 can be obtained from the CP+Jordan representation of $\underline{\mathbf{X}}$ using the method of Ja' Ja' [21].

If, in Cases 3, 5, 8, and 9 of Table 7.1, the optimal GSD solution $\underline{\mathbf{X}}$ has a GSD with $\mathbf{R}_2\mathbf{R}_1^{-1}$ diagonalizable, then the GSD can be transformed into a full CP representation of $\underline{\mathbf{X}}$, and the problems of diverging components do not occur. That is, $\underline{\mathbf{X}}$ is also an optimal solution of the CP problem. The CP representation of $\mathbf{R}_1$ and $\mathbf{R}_2$ can be obtained from the eigendecomposition of $\mathbf{R}_2\mathbf{R}_1^{-1}$ analogous to (2.2). Premultiplying by $\mathbf{Q}_a$ and postmultiplying by $\mathbf{Q}_b^T$ then yields the full CP representation for $\mathbf{X}_1$ and $\mathbf{X}_2$.

7.1. The Jacobi algorithm for the GSD problem of $I \times J \times 2$ arrays. Here, we show how the Jacobi algorithm of De Lathauwer, De Moor, and Vandewalle [6] for solving the GSD problem (6.2) can be modified to the case of $I \times J \times 2$ arrays and all values of R not larger than I and J. That is, the modified Jacobi algorithm can be used to solve the more general class of GSD problems (7.3).

Let $I = J = R$. The Jacobi algorithm of [6] sets out to find $(\mathbf{Q}_a, \mathbf{Q}_b, \mathbf{R}_1, \mathbf{R}_2)$ such that $\mathbf{Q}_a^T\mathbf{Z}_k\mathbf{Q}_b$, $k = 1, 2$ are as upper triangular as possible. Their upper triangular parts are then the estimates of $\mathbf{R}_k$, $k = 1, 2$. The estimates $(\mathbf{Q}_a, \mathbf{Q}_b, \mathbf{R}_1, \mathbf{R}_2)$ are updated by applying Givens rotations to the rows and columns of $\mathbf{Q}_a^T\mathbf{Z}_k\mathbf{Q}_b$, $k = 1, 2$, as follows. Let $\mathbf{G}_{ij}$ be equal to $\mathbf{I}_I$ except for the entries $(\mathbf{G}_{ij})_{ii} = (\mathbf{G}_{ij})_{jj} = \cos\alpha$ and $(\mathbf{G}_{ij})_{ji} = -(\mathbf{G}_{ij})_{ij} = \sin\alpha$, where α is the rotation angle. Let $\widetilde{\mathbf{G}}_{ij}$ be defined as $\mathbf{G}_{ij}$ for a rotation angle β. One sweep of the Jacobi algorithm determines for each (i, j) with $1 \le i < j \le I$, the optimal rotation angles α and β such that $\mathbf{G}_{ij}\mathbf{Q}_a^T\mathbf{Z}_k\mathbf{Q}_b\widetilde{\mathbf{G}}_{ij}^T$, $k = 1, 2$ are as upper triangular as possible. The updated estimates of $(\mathbf{Q}_a, \mathbf{Q}_b, \mathbf{R}_1, \mathbf{R}_2)$ are given by $\mathbf{Q}_a\mathbf{G}_{ij}^T$, $\mathbf{Q}_b\widetilde{\mathbf{G}}_{ij}^T$, and $\mathbf{G}_{ij}\mathbf{R}_k\widetilde{\mathbf{G}}_{ij}^T$, $k = 1, 2$.

Next, consider the general case where possibly $I \neq J$ and $R \le I$, $R \le J$. In the modified Jacobi algorithm, we have the orthonormal variables $\widetilde{\mathbf{Q}}_a$ $(I \times I)$ and $\widetilde{\mathbf{Q}}_b$ $(J \times J)$. The modified Jacobi algorithm maximizes the sum-of-squares of the upper triangular parts of the first R rows and columns of $\widetilde{\mathbf{R}}_k = \widetilde{\mathbf{Q}}_a^T\mathbf{Z}_k\widetilde{\mathbf{Q}}_b$, $k = 1, 2$. These $R \times R$ upper triangular parts are then the estimates of $\mathbf{R}_k$, $k = 1, 2$. The estimates of $\mathbf{Q}_a$ and $\mathbf{Q}_b$ are the first R columns of $\widetilde{\mathbf{Q}}_a$ and $\widetilde{\mathbf{Q}}_b$, respectively. Each sweep of the algorithm consists of two phases. In the first phase, the Givens rotations $\mathbf{G}_{ij}$ $(I \times I)$ and $\widetilde{\mathbf{G}}_{ij}$ $(J \times J)$ are determined as above for each (i, j) with $1 \le i < j \le R$. Within the first R rows and columns of $\widetilde{\mathbf{R}}_k$, $k = 1, 2$, these rotations make the structure as upper triangular as possible.

In the second phase, rotations $\mathbf{G}_i$, $1 \le i \le R$ are determined such that they transfer as much energy as possible from rows $R + 1, \ldots, I$ of $\widetilde{\mathbf{R}}_k$ to row i of (the upper triangular part of) $\widetilde{\mathbf{R}}_k$, $k = 1, 2$. Independently, rotations $\widetilde{\mathbf{G}}_j$, $1 \le i \le R$ are determined such that they transfer as much energy as possible from columns $R + 1, \ldots, J$ of $\widetilde{\mathbf{R}}_k$ to column j of (the upper triangular part of) $\widetilde{\mathbf{R}}_k$, $k = 1, 2$. We first show how to obtain $\mathbf{G}_i$. Let

$$\hat{\mathbf{R}}_i = \begin{bmatrix} (\widetilde{\mathbf{R}}_1)_{ii} & \ldots & (\widetilde{\mathbf{R}}_1)_{iR} & (\widetilde{\mathbf{R}}_2)_{ii} & \ldots & (\widetilde{\mathbf{R}}_2)_{iR} \\ (\widetilde{\mathbf{R}}_1)_{R+1,i} & \ldots & (\widetilde{\mathbf{R}}_1)_{R+1,R} & (\widetilde{\mathbf{R}}_2)_{R+1,i} & \ldots & (\widetilde{\mathbf{R}}_2)_{R+1,R} \\ \vdots & & \vdots & \vdots & & \vdots \\ (\widetilde{\mathbf{R}}_1)_{I,i} & \ldots & (\widetilde{\mathbf{R}}_1)_{I,R} & (\widetilde{\mathbf{R}}_2)_{I,i} & \ldots & (\widetilde{\mathbf{R}}_2)_{I,R} \end{bmatrix} = \mathbf{S}\,\mathbf{D}\,\mathbf{V}^T \tag{7.4}$$

be the singular value decomposition (SVD) of $\hat{\mathbf{R}}_i$. Then $\mathbf{S}^T\hat{\mathbf{R}}_i$ is an orthogonal rotation of the rows of $\hat{\mathbf{R}}_i$ such that its first row has maximum sum-of-squares. The square root of this is equal to the dominant singular value of $\hat{\mathbf{R}}_i$. From $\mathbf{S}^T$, the rotation $\mathbf{G}_i$ can be obtained.

The computation of $\widetilde{\mathbf{G}}_j$ is analogous. Let

$$\bar{\mathbf{R}}_j = \begin{bmatrix} (\widetilde{\mathbf{R}}_1)_{1j} & (\widetilde{\mathbf{R}}_1)_{1,R+1} & \dots & (\widetilde{\mathbf{R}}_1)_{1,J} \\ \vdots & \vdots & & \vdots \\ (\widetilde{\mathbf{R}}_1)_{jj} & (\widetilde{\mathbf{R}}_1)_{j,R+1} & \dots & (\widetilde{\mathbf{R}}_1)_{j,J} \\ (\widetilde{\mathbf{R}}_2)_{1j} & (\widetilde{\mathbf{R}}_2)_{1,R+1} & \dots & (\widetilde{\mathbf{R}}_2)_{1,J} \\ \vdots & \vdots & & \vdots \\ (\widetilde{\mathbf{R}}_2)_{jj} & (\widetilde{\mathbf{R}}_2)_{j,R+1} & \dots & (\widetilde{\mathbf{R}}_2)_{j,J} \end{bmatrix} = \mathbf{S}\,\mathbf{D}\,\mathbf{V}^T \tag{7.5}$$

be the SVD of $\bar{\mathbf{R}}_j$. Then $\bar{\mathbf{R}}_j\mathbf{V}$ is an orthogonal rotation of the columns of $\bar{\mathbf{R}}_j$ such that its first column has maximum sum-of-squares. The square root of this is equal to the dominant singular value of $\bar{\mathbf{R}}_j$. The rotation $\widetilde{\mathbf{G}}_j$ can be obtained from $\mathbf{V}$.

Below, we present the steps of one sweep of the modified Jacobi algorithm.

ONE SWEEP OF THE MODIFIED JACOBI ALGORITHM FOR THE GSD PROBLEM.

Input: $I \times J \times 2$ array $\underline{\mathbf{Z}}$ with $I \times J$ slices $\mathbf{Z}_k$, $k = 1, 2$.
Previous GSD update: $\widetilde{\mathbf{Q}}_a$ $(I \times I)$ and $\widetilde{\mathbf{Q}}_b$ $(J \times J)$ orthonormal, and $\widetilde{\mathbf{R}}_k$ $(I \times J)$, $k = 1, 2$.
Output: New GSD update $\widetilde{\mathbf{Q}}_a$, $\widetilde{\mathbf{Q}}_b$, and $\widetilde{\mathbf{R}}_k$, $k = 1, 2$.

1. (Do for $1 \leq i < j \leq R$.) Let $\mathbf{G}_{ij}$ be equal to $\mathbf{I}_I$ except for the entries $(\mathbf{G}_{ij})_{ii} = (\mathbf{G}_{ij})_{jj} = \cos\alpha$ and $(\mathbf{G}_{ij})_{ji} = -(\mathbf{G}_{ij})_{ij} = \sin\alpha$, where α is the rotation angle. Let $\widetilde{\mathbf{G}}_{ij}$ be equal to $\mathbf{I}_J$ and analogous to $\mathbf{G}_{ij}$ for a rotation angle β. Using the Jacobi algorithm of [6], determine α and β such that the sum-of-squares of the upper triangular part of the first R rows and columns of $\mathbf{G}_{ij}\widetilde{\mathbf{R}}_k\mathbf{G}_{ij}^T$, $k = 1, 2$ are maximal.
   Update $\widetilde{\mathbf{Q}}_a \to \widetilde{\mathbf{Q}}_a\mathbf{G}_{ij}^T$, $\widetilde{\mathbf{Q}}_b \to \widetilde{\mathbf{Q}}_b\widetilde{\mathbf{G}}_{ij}^T$, and $\widetilde{\mathbf{R}}_k \to \mathbf{G}_{ij}\widetilde{\mathbf{R}}_k\widetilde{\mathbf{G}}_{ij}^T$, $k = 1, 2$.
2. (Do for $1 \leq i \leq R$.) Compute the SVD (7.4) and let s_{mn} denote the elements of $\mathbf{S}$. Let $\mathbf{G}_i$ be equal to $\mathbf{I}_I$ except $(\mathbf{G}_i)_{ii} = s_{11}$, $(\mathbf{G}_i)_{i,R+m} = s_{1,m+1}$ for $m = 1, \dots, I-R$, $(\mathbf{G}_i)_{R+m,i} = s_{m+1,1}$ for $m = 1, \dots, I-R$, and $(\mathbf{G}_i)_{mn} = s_{m-R+1,n-R+1}$ for $R+1 \leq m, n \leq I$.
   Update $\widetilde{\mathbf{Q}}_a \to \widetilde{\mathbf{Q}}_a\mathbf{G}_i^T$ and $\widetilde{\mathbf{R}}_k \to \mathbf{G}_i\widetilde{\mathbf{R}}_k$, $k = 1, 2$.
3. (Do for $1 \leq j \leq R$.) Compute the SVD (7.5) and let v_{mn} denote the elements of $\mathbf{V}^T$. Let $\mathbf{G}_j$ be equal to $\mathbf{I}_J$ except $(\mathbf{G}_j)_{jj} = v_{11}$, $(\mathbf{G}_j)_{j,R+m} = v_{1,m+1}$ for $m = 1, \dots, I-R$, $(\mathbf{G}_j)_{R+m,j} = v_{m+1,1}$ for $m = 1, \dots, I-R$, and $(\mathbf{G}_j)_{mn} = v_{m-R+1,n-R+1}$ for $R+1 \leq m, n \leq I$.
   Update $\widetilde{\mathbf{Q}}_b \to \widetilde{\mathbf{Q}}_b\mathbf{G}_j^T$ and $\widetilde{\mathbf{R}}_k \to \widetilde{\mathbf{R}}_k\widetilde{\mathbf{G}}_j^T$, $k = 1, 2$.

8. Numerical experiments. Here, we illustrate the GSD method to avoid diverging CP components for generic $I \times J \times 2$ arrays. For each of the cases in Table 7.1, we randomly generate 50 arrays $\underline{\mathbf{Z}}$ of a chosen size. For each such $\underline{\mathbf{Z}}$, we use the (modified) Jacobi algorithm of De Lathauwer, De Moor, and Vandewall [6] to compute the optimal solution $\underline{\mathbf{X}}$ of the GSD problem (7.3), in terms of its full GSD representation $(\mathbf{Q}_a, \mathbf{Q}_b, \mathbf{R}_1, \mathbf{R}_2)$. The stopping criterion of the (modified) Jacobi algorithm is set to

TABLE 8.1

Types of optimal solutions encountered when solving the GSD problem (7.3) for randomly generated arrays $\underline{\mathbf{Z}}$ in the cases of Table 7.1. For each case, the value of (I, J, R), the number of runs, and the average time to compute the GSD solution and the Jordan form of $\mathbf{R}_2\mathbf{R}_1^{-1}$ (on a Pentium 4 PC) are given. In all runs, the matrix $\mathbf{R}_2\mathbf{R}_1^{-1}$ has distinct eigenvalues $\lambda_1, \ldots, \lambda_p, \mu_1, \ldots, \mu_r$, where λ_j has algebraic multiplicity 1 and μ_i has algebraic multiplicity larger than 1 and geometric multiplicity 1. For each case, the number of solutions with the same (p, r) value are given.

Case, (I, J, R), runs, time	(p,r)	Freq.	(p,r)	Freq.	(p,r)	Freq.	(p,r)	Freq.
Case 2	(0,2)	4	(1,2)	7	(2,3)	2	(4,1)	3
$(I, J, R) = (10, 10, 10)$	(0,3)	2	(1,3)	2	(3,1)	5	(4,2)	1
48 runs, 58 sec.	(0,4)	2	(2,1)	2	(3,2)	4	(5,2)	1
	(1,1)	3	(2,2)	9	(3,3)	1		
Case 3	(0,2)	5	(2,1)	1	(3,2)	5	(6,1)	1
$(I, J, R) = (10, 10, 8)$	(1,1)	8	(2,2)	5	(4,1)	5		
49 runs, 12 sec.	(1,2)	3	(2,3)	1	(4,2)	3		
	(1,3)	3	(3,1)	7	(5,1)	2		
Case 5	(1,1)	2	(2,2)	2	(3,2)	6	(6,1)	15
$(I, J, R) = (10, 10, 8)$	(1,3)	1	(2,3)	1	(4,2)	4	(8,0)	12
50 runs, 32 sec.	(2,1)	1	(3,1)	2	(5,1)	4		
Case 8	(0,2)	4	(1,2)	10	(3,1)	4	(4,2)	3
$(I, J, R) = (10, 8, 8)$	(0,3)	2	(2,1)	1	(3,2)	2	(5,1)	2
50 runs, 10 sec.	(1,1)	3	(2,2)	15	(4,1)	4		
Case 9	(0,2)	4	(1,2)	7	(2,2)	4	(4,1)	9
$(I, J, R) = (10, 8, 6)$	(1,1)	4	(2,1)	5	(3,1)	10	(6,0)	6
49 runs, 3 sec.								

`1e-9`. Next, the Jordan normal form of $\mathbf{R}_2\mathbf{R}_1^{-1}$ is computed, which we denote as $\mathbf{J} = \text{diag}(\lambda_1, \ldots, \lambda_p, \mathbf{J}_{m_1}(\mu_1), \ldots, \mathbf{J}_{m_r}(\mu_r))$, where $\mathbf{J}_{m_j}(\mu_j)$ denotes an $m_j \times m_j$ Jordan block, with $m_j \geq 2$. Numerically, we treat two eigenvalues μ_1 and μ_2 as identical if $|\mu_1 - \mu_2| < 0.01$. The multiple eigenvalue μ is then estimated as the mean of all "identical" eigenvalues. The geometric multiplicity of an eigenvalue μ is determined as the number of singular values s_j of $(\mathbf{R}_2\mathbf{R}_1^{-1} - \mu\,\mathbf{I}_I)$ that satisfy $|s_j| < 0.0001$.

In the (modified) Jacobi algorithm, we use the following initial values for $(\mathbf{Q}_a, \mathbf{Q}_b, \mathbf{R}_1, \mathbf{R}_2)$. In Case 2 in Table 7.1, these are obtained from the "generalized real Schur decomposition" (GRSD) of $\mathbf{Z}_1$ and $\mathbf{Z}_2$, which is computed by means of the QZ-method; see Golub and Van Loan [12]. In the other cases, the slices are first transformed to $\mathbf{U}_R^T\mathbf{Z}_k\mathbf{V}_R$, $k = 1, 2$, where $\mathbf{U}_R$ contains the R dominant left singular vectors of $[\mathbf{Z}_1|\mathbf{Z}_2]$ and $\mathbf{V}_R$ contains the R dominant right singular vectors of $\begin{bmatrix}\mathbf{Z}_1\\ \mathbf{Z}_2\end{bmatrix}$. The initial values are then obtained from $\mathbf{U}_R$, $\mathbf{V}_R$ and the GRSD of $\mathbf{U}_R^T\mathbf{Z}_k\mathbf{V}_R$, $k = 1, 2$.

Table 8.1 summarizes the results of computing the optimal solutions $\underline{\mathbf{X}}$ of the GSD problem (7.3) and the Jordan forms of $\mathbf{R}_2\mathbf{R}_1^{-1}$. As can be seen, a wide variety of values (p, r) is encountered among the optimal solutions $\underline{\mathbf{X}}$. In a few of the 50 runs per case, some identical eigenvalues were not recognized. These runs do not appear in Table 8.1. In all runs in Table 8.1, the estimated eigenvalues $\lambda_1, \ldots, \lambda_p, \mu_1, \ldots, \mu_r$ are distinct. Hence, each μ_i has algebraic multiplicity larger than 1 and geometric multiplicity 1. As observed in Remark 3.2, this is probably due to the fact that the set of these arrays lies dense on the boundary of the set $\mathcal{P}_{(I,J,R)}$. For the solutions with $r \geq 1$, diverging CP components occur, and the GSD of the solution cannot be fully transformed into a CP solution. The solutions with $r = 0$ can be transformed into a nondiverging CP solution, i.e., diverging CP components do not occur.

As can be seen from Table 8.1, all runs in Cases 2, 3, and 8 have a solution with $r \geq 1$. For Case 2, this is in line with the conjecture of Stegeman [40] in Table 7.1. For Cases 3 and 8, this does not seem to support the conjectures of Stegeman [40] in Table 7.1, which state that diverging CP components occur on a set of positive volume (and not almost everywhere). However, trying different values of R in Case 3 yields 15 solutions with $r = 0$ (out of 50) for $R = 4$ and 3 solutions with $r = 0$ (out of 50) for $R = 6$. Hence, it seems that nondiverging CP solutions occur less frequently as R is increased. The same holds for Case 8, where we get 13 solutions with $r = 0$ (out of 50) for $R = J = 4$ and 5 solutions with $r = 0$ (out of 50) for $R = J = 6$. For Cases 5 and 9, there are both solutions with $r = 0$ as well as solutions with $r \geq 1$, which is in line with the conjectures of Stegeman [40] in Table 7.1.

Also listed in Table 8.1 are the average computational times (on a Pentium 4 PC) for the (modified) Jacobi algorithm to terminate and for the computation of the Jordan form of $\mathbf{R}_1\mathbf{R}_1^{-1}$. For Case 2, this is 58 seconds. For comparison, we tried finding an approximate solution to the CP problem (3.2) for random $\underline{\mathbf{Z}}$ as in Case 2, by using the multilinear engine of Paatero [32]. For a convergence criterion of `1e-15` over 1000 consecutive iterations, the algorithm terminated after 40 minutes. However, for the obtained approximate solution $\underline{\mathbf{X}}$, the eigenvalues of $\mathbf{X}_2\mathbf{X}_1^{-1}$ are all clearly distinct. On the other hand, running the Jacobi algorithm on the same $\underline{\mathbf{Z}}$ yields a solution with two groups of identical eigenvalues within 1 minute. Hence, to obtain an equally accurate estimate of the solution $\underline{\mathbf{X}}$ using a CP algorithm requires a very small stopping criterion and takes prohibitively long. This shows the spectacular improvement in efficiency when using the Jacobi GSD algorithm instead.

9. Discussion. We have proposed, analyzed, and demonstrated a method to avoid diverging components when trying to fit the CP model for generic $I \times J \times 2$ arrays and $R \leq I, J$ components. Instead of fitting the CP model, we fit the GSD model. The problems of diverging CP components are likely to occur because the CP model has no optimal solution in these cases. We showed that the GSD model always has an optimal solution. Moreover, the optimal GSD solution is the limit point of the sequence of CP updates, whether it features diverging components or not. Hereby we assume that the GSD model has a unique optimal solution (up to trivial indeterminacies) which is always satisfied in our numerical experiments. Also, we showed that the optimal GSD solution can be represented as the sum of the nondiverging CP components and a sparse Tucker3 part (CP+Jordan form) or as the sum of the nondiverging CP components and a smaller GSD part (CP+GSD form). The CP+Jordan form is essentially unique and sparse. Although it is not an outer-product decomposition, it may still be interpretable to the researcher. From the CP+Jordan representation, we can obtain a rank-revealing decomposition of the optimal GSD solution using the method of Ja' Ja' [21]. However, this decomposition is not essentially unique. The CP+GSD representation is numerically more stable and suitable if only the nondiverging CP components are of interest.

The GSD method not only yields an accurate solution, it is also much faster than trying to fit CP in the case of diverging components. Hence, to compute the CP solution for generic $I \times J \times 2$ arrays, it is advisable to compute the GSD solution instead and then transfer the nondiverging part of the solution into CP components. We may conclude that from a computational as well as a practical point of view, our method is a considerable improvement with respect to facing diverging CP components.

Our analysis is confined to arrays in the sets $\mathcal{R}_I$ in (2.1) and $\mathcal{P}_{(I,J,R)}$ in (6.1). For a given array size, these sets are dense in the space of all arrays. The results

of our numerical experiments and those in Stegeman [38, 40], together with the fact that we consider generic arrays $\underline{\mathbf{Z}}$ to be approximated, lead us to conclude that this confinement is justified in practice. However, from a theoretical point of view, this leaves open the question whether the complement set of $\mathcal{R}_I$ or $\mathcal{P}_{(I,J,R)}$ can contain all best rank-R approximations of a generic $I \times J \times 2$ array.

Stegeman [39] has mathematically analyzed diverging CP components occurring for several generic $I \times J \times 3$ arrays. Whether the SGSD method can also be used for arrays with three slices is currently under investigation.

REFERENCES

[1] R. Bro, *Parafac. Tutorial & applications*, Chemometrics Intell. Lab. Syst., 38 (1997), pp. 149–171.

[2] Y.Z. Cao, Z.P. Chen, C.Y. Mo, H.L. Wu, and R.Q. Yu, *A Parafac algorithm using penalty diagonalization error (PDE) for three-way data array resolution*, The Analyst, 125 (2000), pp. 2303–2310.

[3] J.D. Carroll and J.J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[4] P. Comon, *Independent component analysis, a new concept?*, Signal Process., 36 (1994), pp. 287–314.

[5] L. De Lathauwer, B. De Moor, and J. Vandewalle, *An introduction to independent component analysis*, J. Chemometrics, 14 (2000), pp. 123–149.

[6] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Computation of the canconical decomposition by means of a simultaneous generalized Schur decomposition*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 295–327.

[7] L. De Lathauwer, *A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666.

[8] L. De Lathauwer, *Decompositions of a higher-order tensor in block terms — Part* II: *Definitions and uniqueness*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1033–1066.

[9] L. De Lathauwer and J. Castaing, *Tensor-based techniques for the blind separation of DS-CDMA signals*, Signal Process., 87 (2007), pp. 322–336.

[10] V. De Silva and L.-H. Lim, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1084–1127.

[11] G.H. Golub and J.H. Wilkinson, *Ill-conditioned eigensystems and the computation of the Jordan canonical form*, SIAM Rev., 18 (1976), pp. 578–619.

[12] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., John Hopkins University Press, Baltimore, MD, 1996.

[13] R.A. Harshman, *Foundations of the Parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[14] R.A. Harshman and M.E. Lundy, *Data preprocessing and the extended Parafac model*, in Research Methods for Multimode Data Analysis, H.G. Law, C.W. Snyder Jr., J.A. Hattie, and R.P. McDonald, eds., Praeger, New York, 1984, pp. 216–284.

[15] R.A. Harshman, *The problem and nature of degenerate solutions or decompositions of* 3-*way arrays*, in a talk at the Tensor Decompositions Workshop, Palo Alto, CA, American Institute of Mathematics, 2004.

[16] F.L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp. 164–189.

[17] F.L. Hitchcock, *Multilple invariants and generalized rank of a p-way matrix or tensor*, J. Math. Phys., 7 (1927), pp. 39–70.

[18] P.K. Hopke, P. Paatero, H. Jia, R.T. Ross, and R.A. Harshman, *Three-way (Parafac) factor analysis: Examination and comparison of alternative computational methods as applied to ill-conditioned data*, Chemometrics Intell. Labor. Syst., 43 (1998), pp. 25–42.

[19] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, MA, 1993.

[20] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, Wiley, New York, 2001.

[21] J. Ja' Ja', *Optimal evaluation of pairs of bilinear forms*, SIAM J. Comput., 8 (1979), pp. 443–462.

[22] T. Jiang and N.D. Sidiropoulos, *Kruskal's permutation lemma and the identification of Candecomp/Parafac and bilinear models with constant modulus constraints*, IEEE Trans. Signal Process., 52 (2004), pp. 2625–2636.

[23] W.P. Krijnen, *Convergence of the sequence of parameters generated by alternating least squares algorithms*, Comput. Statist. Data Anal., 51 (2006), pp. 481–489.

[24] W.P. Krijnen, T.K. Dijkstra, and A. Stegeman, *On the non-existence of optimal solutions and the occurrence of "degeneracy" in the Candecomp/Parafac model*, Psychometrika, 73 (2008), pp. 431–439.

[25] P.M. Kroonenberg, *Applied Multiway Data Analysis*, Wiley Ser. Probab. Stat., Wiley Interscience, New York, 2008.

[26] J.B. Kruskal, *Three-way arrays: Rank and uniqueness of trilinear decompositions, with applications to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95–138.

[27] J.B. Kruskal, R.A. Harshman, and M.E. Lundy, *How 3-MFA data can cause degenerate Parafac solutions, among other relationships*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North-Holland, Amsterdam, 1989, pp. 115–121.

[28] S.E. Leurgans, R.T. Ross, and R.B. Abel, *A decomposition for three-way arrays*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1064–1083.

[29] L.-H. Lim, *Optimal solutions to non-negative Parafac/multilinear NMF always exist*, in a talk at the Workshop on Tensor Decompositions and Applications, CIRM, Luminy, Marseille, France, 2005.

[30] B.C. Mitchell and D.S. Burdick, *Slowly converging Parafac sequences: Swamps and two-factor degeneracies*, J. Chemometrics, 8 (1994), pp. 155–168.

[31] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, San Diego, 1970.

[32] P. Paatero, *The multilinear engine—A table-driven least squares program for solving multilinear problems, including the n-way parallel factor analysis model*, J. Comput. Graph. Statist., 8 (1999), pp. 854–888.

[33] P. Paatero, *Construction and analysis of degenerate Parafac models*, J. Chemometrics, 14 (2000), pp. 285–299.

[34] W.S. Rayens and B.C. Mitchell, *Two-factor degeneracies and a stabilization of Parafac*, Chemometrics Intell. Labor. Syst., 38 (1997), pp. 173–181.

[35] N. Sidiropoulos, G. Giannakis, and R. Bro, *Blind Parafac receivers for DS-CDMA systems*, IEEE Trans. Signal Process., 48 (2000), pp. 810–823.

[36] N. Sidiropoulos, R. Bro, and G. Giannakis, *Parallel factor analysis in sensor array processing*, IEEE Trans. Signal Process., 48 (2000), pp. 2377–2388.

[37] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis: Applications in the Chemical Sciences*, Wiley, New York, 2004.

[38] A. Stegeman, *Degeneracy in Candecomp/Parafac explained for $p \times p \times 2$ arrays of rank $p+1$ or higher*, Psychometrika, 71 (2006), pp. 483–501.

[39] A. Stegeman, *Degeneracy in Candecomp/Parafac explained for several three-sliced arrays with a two-valued typical rank*, Psychometrika, 72 (2007), pp. 601–619.

[40] A. Stegeman, *Low-rank approximation of generic $p \times q \times 2$ arrays and diverging components in the Candecomp/Parafac model*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 988–1007.

[41] A. Stegeman, J.M.F. Ten Berge, and L. De Lathauwer, *Sufficient conditions for uniqueness in Candecomp/Parafac and Indscal with random component matrices*, Psychometrika, 71 (2006), pp. 219–229.

[42] A. Stegeman and N.D. Sidiropoulos, *On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition*, Linear Algebra Appl., 420 (2007), pp. 540–552.

[43] J.M.F Ten Berge, H.A.L. Kiers, and J. De Leeuw, *Explicit Candecomp/Parafac solutions for a contrived $2 \times 2 \times 2$ array of rank three*, Psychometrika, 53 (1988), pp. 579–584.

[44] J.M.F. Ten Berge and H.A.L. Kiers, *Simplicity of core arrays in three-way principal component analysis and the typical rank of $p \times q \times 2$ arrays*, Linear Algebra Appl., 294 (1999), pp. 169–179.

[45] G. Tomasi and R. Bro, *A comparison of algorithms for fitting the Parafac model*, Comput. Statist. Data Anal., 50 (2006), pp. 1700–1734.

[46] L.R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[47] A.-J. Van der Veen and A. Paulraj, *An analytical constant modulus algorithm*, IEEE Trans. Signal Process., 44 (1996), pp. 1136–1155.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1639–1657

COMPUTING THE FRÉCHET DERIVATIVE OF THE MATRIX EXPONENTIAL, WITH AN APPLICATION TO CONDITION NUMBER ESTIMATION*

AWAD H. AL-MOHY† AND NICHOLAS J. HIGHAM†

Abstract. The matrix exponential is a much-studied matrix function having many applications. The Fréchet derivative of the matrix exponential describes the first-order sensitivity of e^A to perturbations in A and its norm determines a condition number for e^A. Among the numerous methods for computing e^A the scaling and squaring method is the most widely used. We show that the implementation of the method in [N. J. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1179–1193] can be extended to compute both e^A and the Fréchet derivative at A in the direction E, denoted by $L(A, E)$, at a cost about three times that for computing e^A alone. The algorithm is derived from the scaling and squaring method by differentiating the Padé approximants and the squaring recurrence, reusing quantities computed during the evaluation of the Padé approximant, and intertwining the recurrences in the squaring phase. To guide the choice of algorithmic parameters, an extension of the existing backward error analysis for the scaling and squaring method is developed which shows that, modulo rounding errors, the approximations obtained are $e^{A+\Delta A}$ and $L(A + \Delta A, E + \Delta E)$, with the same ΔA in both cases, and with computable bounds on $\|\Delta A\|$ and $\|\Delta E\|$. The algorithm for $L(A, E)$ is used to develop an algorithm that computes e^A together with an estimate of its condition number. In addition to results specific to the exponential, we develop some results and techniques for arbitrary functions. We show how a matrix iteration for $f(A)$ yields an iteration for the Fréchet derivative and show how to efficiently compute the Fréchet derivative of a power series. We also show that a matrix polynomial and its Fréchet derivative can be evaluated at a cost at most three times that of computing the polynomial itself and give a general framework for evaluating a matrix function and its Fréchet derivative via Padé approximation.

Key words. matrix function, Fréchet derivative, matrix polynomial, matrix iteration, matrix exponential, condition number estimation, scaling and squaring method, Padé approximation, backward error analysis

AMS subject classifications. 15A60, 65F30

DOI. 10.1137/080716426

1. Introduction.

The sensitivity of a matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ to small perturbations is governed by the Fréchet derivative. The Fréchet derivative at a point $A \in \mathbb{C}^{n\times n}$ is a linear mapping

$$\begin{array}{rcl} \mathbb{C}^{n\times n} & \stackrel{L(A)}{\longrightarrow} & \mathbb{C}^{n\times n} \\ E & \longmapsto & L(A, E) \end{array}$$

such that for all $E \in \mathbb{C}^{n\times n}$,

$$f(A + E) - f(A) - L(A, E) = o(\|E\|), \tag{1.1}$$

and it therefore describes the first-order effect on f of perturbations in A. If we need to show the dependence of L on f we will write $L_f(A, E)$.

*Received by the editors February 22, 2008; accepted for publication (in revised form) by M. H. Gutknecht August 26, 2008; published electronically January 23, 2009.
http://www.siam.org/journals/simax/30-4/71642.html

†School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK (almohy@maths.manchester.ac.uk, http://www.maths.manchester.ac.uk/~almohy, higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham). The work of the second author was supported by a Royal Society-Wolfson Research Merit Award and by Engineering and Physical Sciences Research Council grant EP/D079403.

It is desirable to be able to evaluate efficiently both $f(A)$ and the Fréchet derivative in order to obtain sensitivity information or to apply an optimization algorithm requiring derivatives. However, while the numerical computation of matrix functions is quite well developed, fewer methods are available for the Fréchet derivative, and the existing methods for $L(A,E)$ usually do not fully exploit the fact that $f(A)$ is being computed [6].

The norm of the Fréchet derivative yields a condition number [6, Theorem 3.1]:

$$\operatorname{cond}(f,A) := \lim_{\epsilon\to 0} \sup_{\|E\|\le\epsilon\|A\|} \frac{\|f(A+E)-f(A)\|}{\epsilon\|f(A)\|} = \frac{\|L(A)\|\,\|A\|}{\|f(A)\|}, \tag{1.2}$$

where

$$\|L(A)\| := \max_{Z\neq 0} \frac{\|L(A,Z)\|}{\|Z\|} \tag{1.3}$$

and the norm is any matrix norm. When evaluating $f(A)$ we would like to be able to efficiently estimate $\operatorname{cond}(f,A)$; (1.3) shows that to do so we need to approximately maximize the norm of $L(A,Z)$ over all Z of unit norm.

The main aim of this work is to develop an efficient algorithm for simultaneously computing e^A and $L(A,E)$ and to use it to construct an algorithm for computing e^A along with an estimate of $\operatorname{cond}(\exp,A)$. The need for such algorithms is demonstrated by a recent paper in econometrics [8] in which the authors state that "One problem we did discover, that has not been accentuated in the literature, is that altering the stability properties of the coefficient matrix through a change in just one parameter can dramatically alter the theoretical and computed matrix exponential." If $A = A(t)$ depends smoothly on a vector $t \in \mathbb{C}^p$ of parameters then the change in e^A induced by small changes θh in t ($\theta \in \mathbb{C}$, $h \in \mathbb{C}^p$) is approximated by $\theta L(A, \sum_{i=1}^p h_i \partial A(t)/\partial t_i)$, since

$$\begin{aligned} f(A(t+\theta h)) &= f\left(A + \theta \sum_{i=1}^p \frac{\partial A(t)}{\partial t_i} h_i + O(\theta^2)\right) \\ &= f(A) + L\left(A, \theta \sum_{i=1}^p \frac{\partial A(t)}{\partial t_i} h_i + O(\theta^2)\right) + o(\theta) \\ &= f(A) + \theta L\left(A, \sum_{i=1}^p \frac{\partial A(t)}{\partial t_i} h_i\right) + o(\theta). \end{aligned}$$

Thus a single Fréchet derivative evaluation with $h = e_j$ (the jth unit vector) provides the information that the authors of [8] needed about the effect of changing a single parameter t_j.

We begin in section 2 by recalling a useful connection between the Fréchet derivative of a function and the same function evaluated at a certain block triangular matrix. We illustrate how this relation can be used to derive new iterations for computing $L(A,E)$ given an iteration for $f(A)$. Then in section 3 we show how to efficiently evaluate the Fréchet derivative when f has a power series expansion, by exploiting a convenient recurrence for the Fréchet derivative of a monomial. In section 4 we show that under reasonable assumptions a matrix polynomial and its Fréchet derivative can both be evaluated at a cost at most three times that of evaluating the polynomial itself. Then in section 5 we show how to evaluate the Fréchet derivative of a

rational function and give a framework for evaluating f and its Fréchet derivative via Padé approximants. In section 6 we apply this framework to the scaling and squaring algorithm for e^A [14], [17], and in particular to the implementation of Higham [5], which is the basis of MATLAB's `expm` function. We extend Higham's analysis to show that, modulo rounding errors, the approximations obtained from the new algorithm are $e^{A+\Delta A}$ and $L(A+\Delta A, E+\Delta E)$, with the *same* ΔA in both cases—a genuine backward error result. The computable bounds on $\|\Delta A\|$ and $\|\Delta E\|$ enable us to choose the algorithmic parameters in an optimal fashion. The new algorithm is shown to have significant advantages over existing ones. In section 7 we combine the new algorithm for $L(A,E)$ with an existing matrix 1-norm estimator to develop an algorithm for computing both e^A and an estimate of its condition number, and we show experimentally that the condition estimate can provide a useful guide to the accuracy of the scaling and squaring algorithm. Some concluding remarks are given in section 8.

2. Fréchet derivative via function of block triangular matrix. The following result shows that the Fréchet derivative appears as the $(1,2)$ block when f is evaluated at a certain block triangular matrix. Let $\mathcal{D}$ denote an open subset of $\mathbb{R}$ or $\mathbb{C}$.

THEOREM 2.1. *Let f be $2n-1$ times continuously differentiable on $\mathcal{D}$ and let the spectrum of X lie in $\mathcal{D}$. Then*

$$f\left(\begin{bmatrix} X & E \\ 0 & X \end{bmatrix}\right) = \begin{bmatrix} f(X) & L(X,E) \\ 0 & f(X) \end{bmatrix}. \tag{2.1}$$

Proof. See Mathias [13, Theorem 2.1] or Higham [6, section 3.1]. The result is also proved by Najfeld and Havel [15, Theorem 4.11] under the assumption that f is analytic. ∎

The significance of Theorem 2.1 is that given a smooth enough f and any method for computing $f(A)$, we can compute the Fréchet derivative by applying the method to the $2n \times 2n$ matrix in (2.1). The doubling in size of the problem is unwelcome, but if we exploit the block structure the computational cost can be reduced. Moreover, the theorem can provide a simple means to derive, and prove the convergence of, iterations for computing the Fréchet derivative.

To illustrate the use of the theorem we consider the principal square root function, $f(A) = A^{1/2}$, which for $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on $\mathbb{R}^-$ (the closed negative real axis) is the unique square root X of A whose spectrum lies in the open right half-plane. The Denman–Beavers iteration

$$\begin{aligned} X_{k+1} &= \frac{1}{2}\left(X_k + Y_k^{-1}\right), \qquad X_0 = A, \\ Y_{k+1} &= \frac{1}{2}\left(Y_k + X_k^{-1}\right), \qquad Y_0 = I \end{aligned} \tag{2.2}$$

is a Newton variant that converges quadratically with [6, section 6.3]

$$\lim_{k\to\infty} X_k = A^{1/2}, \qquad \lim_{k\to\infty} Y_k = A^{-1/2}. \tag{2.3}$$

It is easy to show that if we apply the iteration to $\widetilde{A} = \left[\begin{smallmatrix} A & E \\ 0 & A \end{smallmatrix}\right]$ then iterates $\widetilde{X}_k$ and $\widetilde{Y}_k$ are produced for which

$$\widetilde{X}_k = \begin{bmatrix} X_k & F_k \\ 0 & X_k \end{bmatrix}, \qquad \widetilde{Y}_k = \begin{bmatrix} Y_k & G_k \\ 0 & Y_k \end{bmatrix},$$

where

$$\begin{aligned} F_{k+1} &= \frac{1}{2}\left(F_k - Y_k^{-1} G_k Y_k^{-1}\right), \qquad F_0 = E, \\ G_{k+1} &= \frac{1}{2}\left(G_k - X_k^{-1} F_k X_k^{-1}\right), \qquad G_0 = 0. \end{aligned} \tag{2.4}$$

By applying (2.3) and Theorem 2.1 to $\widetilde{A}$ we conclude that

$$\lim_{k\to\infty} F_k = L_{x^{1/2}}(A,E), \qquad \lim_{k\to\infty} G_k = L_{x^{-1/2}}(A,E). \tag{2.5}$$

Moreover, scaling strategies for accelerating the convergence of (2.2) [6, section 6.5] yield corresponding strategies for (2.4).

The next result shows quite generally that differentiating a fixed point iteration for a matrix function yields a fixed point iteration for the Fréchet derivative.

THEOREM 2.2. *Let f and g be $n-1$ times continuously differentiable on $\mathcal{D}$. Suppose that for any matrix $X \in \mathbb{C}^{n\times n}$ whose spectrum lies in $\mathcal{D}$, g has the fixed point $f(X)$, that is, $f(X) = g(f(X))$. Then for any such X, L_g at $f(X)$ has the fixed point $L_f(X,E)$ for all E.*

Proof. Applying the chain rule to $f(X) \equiv g(f(X))$ gives the relation $L_f(X,E) = L_g(f(X), L_f(X,E))$, which is the result. ∎

The iteration (2.4) for computing the Fréchet derivative of the square root function is new, and other new iterations for the Fréchet derivative of the matrix square root and related functions can be derived, and their convergence proved, in the same way, or directly by using Theorem 2.2. In the case of the Newton iteration for the matrix sign function this approach yields an iteration for the Fréchet derivative proposed by Kenney and Laub [10, Theorem 3.3] (see also [6, Theorem 5.7]) and derived using Theorem 2.1 by Mathias [13].

In the rest of this paper we consider the situation in which the underlying method for computing $f(A)$ is based on direct approximation rather than iteration, and we develop techniques that are more sophisticated than a direct application of Theorem 2.1.

3. Fréchet derivative via power series. When f has a power series expansion the Fréchet derivative can be expressed as a related series expansion.

THEOREM 3.1. *Suppose f has the power series expansion $f(x) = \sum_{k=0}^{\infty} a_k x^k$ with radius of convergence r. Then for $A, E \in \mathbb{C}^{n\times n}$ with $\|A\| < r$, the Fréchet derivative*

$$L_f(A,E) = \sum_{k=1}^{\infty} a_k \sum_{j=1}^{k} A^{j-1} E A^{k-j}. \tag{3.1}$$

Proof. See [6, Problem 3.6]. ∎

The next theorem gives a recurrence that can be used to evaluate (3.1).

THEOREM 3.2. *Under the assumptions of Theorem* 3.1*,*

$$L_f(A,E) = \sum_{k=1}^{\infty} a_k M_k, \tag{3.2}$$

where $M_k = L_{x^k}(A,E)$ satisfies the recurrence

$$M_k = M_{\ell_1} A^{\ell_2} + A^{\ell_1} M_{\ell_2}, \qquad M_1 = E, \tag{3.3}$$

with $k = \ell_1 + \ell_2$ and ℓ_1 and ℓ_2 positive integers. In particular,

$$M_k = M_{k-1}A + A^{k-1}M_1, \qquad M_1 = E. \tag{3.4}$$

In addition,

$$\|f(A)\| \le \widetilde{f}(\|A\|), \qquad \|L_f(A)\| \le \widetilde{f}'(\|A\|), \tag{3.5}$$

where $\widetilde{f}(x) = \sum_{k=0}^{\infty} |a_k| x^k$.

Proof. Since the power series can be differentiated term-by-term within its radius of convergence, we have

$$L_f(A,E) = \sum_{k=1}^{\infty} a_k M_k, \qquad M_k = L_{x^k}(A,E).$$

One way to develop the recurrence (3.3) is by applying Theorem 2.1 to the monomial $x^k = x^{\ell_1+\ell_2}$. A more direct approach is to use the product rule for Fréchet derivatives [6, Theorem 3.3] to obtain

$$M_k = L_{x^k}(A,E) = L_{x^{\ell_1}}(A,E)A^{\ell_2} + A^{\ell_1}L_{x^{\ell_2}}(A,E) = M_{\ell_1}A^{\ell_2} + A^{\ell_1}M_{\ell_2}.$$

Taking $\ell_1 = k-1$ and $\ell_2 = 1$ gives (3.4). It is straightforward to see that $\|f(A)\| \le \widetilde{f}(\|A\|)$. Taking norms in (3.1) gives

$$\|L_f(A,E)\| \le \|E\| \sum_{k=1}^{\infty} k|a_k| \|A\|^{k-1} = \|E\| \widetilde{f}'(\|A\|),$$

and maximizing over all nonzero E gives $\|L_f(A)\| \le \widetilde{f}'(\|A\|)$. ∎

The recurrence (3.3) will prove very useful in the rest of the paper.

4. Cost analysis for polynomials. Practical methods for approximating $f(A)$ may truncate a Taylor series to a polynomial or use a rational approximation. Both cases lead to the need to evaluate both a polynomial and its Fréchet derivative at the same argument. The question arises "what is the extra cost of computing the Fréchet derivative?" Theorem 3.2 does not necessarily answer this question because it only describes one family of recurrences for evaluating the Fréchet derivative. Moreover, the most efficient polynomial evaluation schemes are based on algebraic rearrangements that avoid explicitly forming all the matrix powers. Does an efficient evaluation scheme for a polynomial p also yield an efficient evaluation scheme for L_p?

Consider schemes for evaluating $p_m(X)$, where p_m is a polynomial of degree m and $X \in \mathbb{C}^{n\times n}$, that consist of s steps of the form

$$q_1^{(k)}(X) = q_2^{(k-1)}(X)\, q_3^{(k-1)}(X) + q_4^{(k-1)}(X), \qquad k = 1\colon s, \tag{4.1a}$$

$$\deg q_i^{(k)} < m, \quad i = 1\colon 4,\ k < s, \qquad \deg q_i^{(k)} \ge 1, \quad i = 2\colon 3, \tag{4.1b}$$

where the q_i are polynomials, $q_i^{(k)}$, $i = 2\colon 4$, is a linear combination of $q_1^{(1)}, \dots, q_1^{(k-1)}$, and $p_m(X) = q_1^{(s)}(X)$. This class contains all schemes of practical interest, which include Horner's method, evaluation by explicit powers, and the Paterson and Stockmeyer method [16] (all of which are described in [6, section 4.2]), as well as more ad hoc schemes such as those described below. We measure the cost of the scheme by the

number of matrix multiplications it requires. The next result shows that the overhead of evaluating the Fréchet derivative is at most twice the original cost.

THEOREM 4.1. *Let p be a polynomial and let π_p denote the cost of evaluating $p(X)$ by any scheme of the form* (4.1). *Let σ_p denote the extra cost required to compute $L_p(X,E)$ by using the scheme obtained by differentiating the scheme for $p(X)$. Then $\sigma_p \le 2\pi_p$.*

Proof. The proof is by induction on the degree m of the polynomial. For $m = 1$, $p_1(x) = b_0 + b_1 x$ and the only possible scheme is the obvious evaluation $p_1(X) = b_0 I + b_1 X$ with $\pi_1 = 0$. The corresponding Fréchet derivative scheme is $L_{p_1}(X,E) = b_1 E$ and $\sigma_1 = 0$, so the result is trivially true for $m = 1$. Suppose the result is true for all polynomials of degree at most $m-1$ and consider a polynomial p_m of degree m. By (4.1) the last stage of the scheme can be written $p_m(X) = q_2^{(s-1)}(X) q_3^{(s-1)}(X) + q_4^{(s-1)}(X)$, where the polynomials $q_i \equiv q_i^{(s-1)}$, $i = 2\colon 4$ are all of degree less than m. Note that $\pi_{p_m} = \pi_{q_2} + \pi_{q_3} + \pi_{q_4} + 1$ and by the inductive hypothesis, $\sigma_{q_i} \le 2\pi_{q_i}$, $i = 2\colon 4$. Now $L_{p_m}(X,E) = L_{q_2}(X,E) q_3(X) + q_2(X) L_{q_3}(X,E) + L_{q_4}(X,E)$ by the product rule and so

$$\sigma_{p_m} \le \sigma_{q_2} + \sigma_{q_3} + \sigma_{q_4} + 2 \le 2(\pi_{q_2} + \pi_{q_3} + \pi_{q_4} + 1) = 2\pi_{p_m},$$

as required. This proof tacitly assumes that there are no dependencies between the $q_i^{(k)}$ that reduce the cost of evaluating p, for example, $q_2^{(s-1)} = q_3^{(s-1)}$. However, any dependencies equally benefit the L_p evaluation and the result remains valid. ∎

To illustrate the theorem, consider the polynomial $p(X) = I + X + X^2 + X^3 + X^4 + X^5$. Rewriting it as

$$p(X) = I + X\left(I + X^2 + X^4\right) + X^2 + X^4,$$

we see that $p(X)$ can be evaluated in just three multiplications via $X_2 = X^2$, $X_4 = X_2^2$, and $p(X) = I + X(I + X_2 + X_4) + X_2 + X_4$. Differentiating gives

$$\begin{aligned} L_p(X,E) &= L_{x(1+x^2+x^4)}(X,E) + M_2 + M_4 \\ &= E(I + X_2 + X_4) + X(M_2 + M_4) + M_2 + M_4, \end{aligned}$$

where $M_2 = XE + EX$ and $M_4 = M_2 X_2 + X_2 M_2$ by (3.3). Hence the Fréchet derivative can be evaluated with six additional multiplications, and the total cost is nine multiplications.

5. Computational framework. For a number of important functions f, such as the exponential, the logarithm, and the sine and cosine, successful algorithms for $f(A)$ have been built on the use of Padé approximants: a Padé approximant r_m of f of suitable degree m is evaluated at a transformed version of A and the transformation is then undone. Here, $r_m(x) = p_m(x)/q_m(x)$ with p_m and q_m polynomials of degree m such that $f(x) - r_m(x) = O(x^{2m+1})$ [2]. It is natural to make use of this Padé approximant by approximating L_f by the Fréchet derivative L_{r_m} of r_m. The next result shows how to evaluate L_{r_m}.

LEMMA 5.1. *The Fréchet derivative L_{r_m} of the rational function $r_m(x) = p_m(x)/q_m(x)$ satisfies*

$$q_m(A) L_{r_m}(A,E) = L_{p_m}(A,E) - L_{q_m}(A,E) r_m(A). \tag{5.1}$$

Proof. Applying the Fréchet derivative product rule to $q_m r_m = p_m$ gives

$$L_{p_m}(A,E) = L_{q_m r_m}(A,E) = L_{q_m}(A,E)r_m(A) + q_m(A)L_{r_m}(A,E),$$

which rearranges to the result. ∎

We can now state a general framework for simultaneously approximating $f(A)$ and $L_f(A,E)$ in a way that reuses matrix multiplications from the approximation of f in the approximation of L_f.

1. Choose a suitable Padé degree m and transformation function g and set $A \leftarrow g(A)$.
2. Devise efficient schemes for evaluating $p_m(A)$ and $q_m(A)$.
3. Fréchet differentiate the schemes in the previous step to obtain schemes for evaluating $L_{p_m}(A,E)$ and $L_{q_m}(A,E)$. Use the recurrences (3.3) and (3.4) as necessary.
4. Solve $q_m(A)r_m(A) = p_m(A)$ for $r_m(A)$.
5. Solve $q_m(A)L_{r_m}(A,E) = L_{p_m}(A,E) - L_{q_m}(A,E)r_m(A)$ for $L_{r_m}(A,E)$.
6. Apply the appropriate transformations to $r_m(A)$ and $L_{r_m}(A,E)$ that undo the effect of the initial transformation on A.

In view of Theorem 4.1, the cost of this procedure is at most $(3\pi_m + 1)M + 2D$, where $\pi_m M$ is the cost of evaluating both $p_m(A)$ and $q_m(A)$, and M and D denote a matrix multiplication and the solution of a matrix equation, respectively.

If we are adding the capability to approximate the Fréchet derivative to an existing Padé-based method for $f(A)$ then our attention will focus on step 1, where we must reconsider the choice of m and transformation to ensure that both f and L_f are approximated to sufficient accuracy.

In the next section we apply this framework to the matrix exponential.

6. Scaling and squaring algorithm for the exponential and its Fréchet derivative. The scaling and squaring method for computing the exponential of $A \in \mathbb{C}^{n\times n}$ is based on the relation

$$e^A = \left(e^{2^{-s}A}\right)^{2^s}. \tag{6.1}$$

For suitably chosen nonnegative integers s and m, this method approximates $e^{2^{-s}A}$ by $r_m(2^{-s}A)$, where r_m is the $[m/m]$ Padé approximant to e^x, and it takes $e^A \approx (r_m(2^{-s}A))^{2^s}$. A choice of the parameters s and m with a certain optimality property is given in the following algorithm from [5], [6, Algorithm 10.20], which forms the basis of MATLAB's `expm` function.

ALGORITHM 6.1 (scaling and squaring algorithm for exponential). *This algorithm evaluates the matrix exponential $X = e^A$ of $A \in \mathbb{C}^{n\times n}$ using the scaling and squaring method. It uses the parameters θ_m given in Table* 6.1. *The algorithm is intended for IEEE double precision arithmetic.*

```
1  for m = [3 5 7 9]
2      if ||A||_1 ≤ θ_m, evaluate X = r_m(A) using (6.11) and (6.14), quit, end
3  end
4  A ← 2^{-s}A with s = ⌈log_2(||A||_1/θ_13)⌉
5  Evaluate r_13(A) using (6.14) and the preceding equations.
6  X = r_13(A)^{2^s} by repeated squaring.
```

Cost: $(\pi_m + s)M + D$, where m is the degree of Padé approximant used and π_m (tabulated in [5, Table 2.2]) is the cost of evaluating p_m and q_m.

Our aim is now to adapt this algorithm to compute $L_{\exp}(A,E)$ along with e^A.

TABLE 6.1
Maximal values ℓ_m of $\|2^{-s}A\|$ such that the backward error bound (6.10) *does not exceed $u = 2^{-53}$, along with maximal values θ_m such that a bound for $\|\Delta A\|/\|A\|$ does not exceed u.*

m	1	2	3	4	5	6	7	8	9	10
θ_m	3.65e-8	5.32e-4	1.50e-2	8.54e-2	2.54e-1	5.41e-1	9.50e-1	1.47e0	2.10e0	2.81e0
ℓ_m	2.11e-8	3.56e-4	1.08e-2	6.49e-2	2.00e-1	4.37e-1	7.83e-1	1.23e0	1.78e0	2.42e0

m	11	12	13	14	15	16	17	18	19	20
θ_m	3.60e0	4.46e0	5.37e0	6.33e0	7.34e0	8.37e0	9.44e0	1.05e1	1.17e1	1.28e1
ℓ_m	3.13e0	3.90e0	4.74e0	5.63e0	6.56e0	7.52e0	8.53e0	9.56e0	1.06e1	1.17e1

A recurrence for the Fréchet derivative of the exponential can be obtained by differentiating (6.1). Note first that differentiating the identity $e^A = (e^{A/2})^2$ using the chain rule [6, Theorem 3.4] along with $L_{x^2}(A,E) = AE + EA$ gives the relation

$$\begin{aligned} L_{\exp}(A,E) &= L_{x^2}\big(e^{A/2}, L_{\exp}(A/2,E/2)\big) \\ &= e^{A/2}L_{\exp}(A/2,E/2) + L_{\exp}(A/2,E/2)e^{A/2}. \end{aligned} \tag{6.2}$$

Repeated use of this relation leads to the recurrence

$$\begin{aligned} \widetilde{L}_s &= L_{\exp}(2^{-s}A, 2^{-s}E), \\ \widetilde{L}_{i-1} &= e^{2^{-i}A}\widetilde{L}_i + \widetilde{L}_i e^{2^{-i}A}, \qquad i = s\colon -1\colon 1 \end{aligned} \tag{6.3}$$

for $\widetilde{L}_0 = L_{\exp}(A,E)$. Our numerical method replaces $L_{\exp}$ by L_{r_m} and $e^{2^{-i}A}$ by $r_m(2^{-s}A)^{2^{s-i}}$, producing approximations L_i to $\widetilde{L}_i$:

$$\begin{aligned} X_s &= r_m(2^{-s}A), \\ L_s &= L_{r_m}(2^{-s}A, 2^{-s}E), \\ \left.\begin{aligned} L_{i-1} &= X_i L_i + L_i X_i \\ X_{i-1} &= X_i^2 \end{aligned}\right\} & \quad i = s\colon -1\colon 1. \end{aligned} \tag{6.4}$$

The key question is what can be said about the accuracy or stability of L_0 relative to that of the approximation $X_0 = (r_m(2^{-s}A))^{2^s}$ to e^A. To answer this question we recall the key part of the error analysis from [5] (see also [6, section 10.3]), which is summarized in the following result. We denote by log the principal logarithm of $A \in \mathbb{C}^{n\times n}$, which is defined for A with no eigenvalues on $\mathbb{R}^-$ and is the unique logarithm whose eigenvalues all have imaginary parts in $(-\pi,\pi)$.

THEOREM 6.2. *Suppose that*

$$\|e^{-A}r_m(A) - I\| < 1, \qquad \|A\| < \min\{\,|t| : q_m(t) = 0\,\} \tag{6.5}$$

for some consistent matrix norm, so that $g_m(A) = \log(e^{-A}r_m(A))$ is guaranteed to be defined. Then $r_m(A) = e^{A+g_m(A)}$ and $\|g_m(A)\| \le -\log(1 - \|e^{-A}r_m(A) - I\|)$. In particular, if (6.5) *is satisfied with $A \leftarrow 2^{-s}A$ then*

$$r_m(2^{-s}A) = e^{2^{-s}A + g_m(2^{-s}A)}, \tag{6.6}$$

so that $r_m(2^{-s}A)^{2^s} = e^{A + 2^s g_m(2^{-s}A)}$, where

$$\frac{\|2^s g_m(2^{-s}A)\|}{\|A\|} \le \frac{-\log(1 - \|e^{-2^{-s}A}r_m(2^{-s}A) - I\|)}{\|2^{-s}A\|}. \tag{6.7}$$

∎

Differentiating (6.6) gives, using the chain rule,

$$\begin{aligned} L_s &= L_{r_m}(2^{-s}A, 2^{-s}E) \\ &= L_{\exp}\bigl(2^{-s}A + g_m(2^{-s}A), 2^{-s}E + L_{g_m}(2^{-s}A, 2^{-s}E)\bigr). \end{aligned} \tag{6.8}$$

From (6.4), (6.6), and (6.8),

$$\begin{aligned} L_{s-1} &= r_m(2^{-s}A)L_s + L_s r_m(2^{-s}A) \\ &= e^{2^{-s}A+g_m(2^{-s}A)}\, L_{\exp}\bigl(2^{-s}A + g_m(2^{-s}A), 2^{-s}E + L_{g_m}(2^{-s}A, 2^{-s}E)\bigr) \\ &\quad + L_{\exp}\bigl(2^{-s}A + g_m(2^{-s}A), 2^{-s}E + L_{g_m}(2^{-s}A, 2^{-s}E)\bigr)\, e^{2^{-s}A+g_m(2^{-s}A)} \\ &= L_{\exp}\bigl(2^{-(s-1)}A + 2g_m(2^{-s}A), 2^{-(s-1)}E + L_{g_m}(2^{-s}A, 2^{-(s-1)}E)\bigr), \end{aligned}$$

where we have used (6.2) and the fact that L is linear in its second argument. Continuing this argument inductively, and using

$$X_i = X_s^{2^{s-i}} = \bigl(e^{2^{-s}A+g_m(2^{-s}A)}\bigr)^{2^{s-i}} = e^{2^{-i}A+2^{s-i}g_m(2^{-s}A)},$$

we obtain the following result.

THEOREM 6.3. *If* (6.5) *is satisfied with* $A \leftarrow 2^{-s}A$ *then* L_0 *from* (6.4) *satisfies*

$$L_0 = L_{\exp}\bigl(A + 2^s g_m(2^{-s}A), E + L_{g_m}(2^{-s}A, E)\bigr). \tag{6.9}$$ □

Theorem 6.3 is a backward error result: it says that L_0 is the exact Fréchet derivative for the exponential of a perturbed matrix in a perturbed direction. We emphasize that the backward error is with respect to the effect of truncation errors in the Padé approximation, not to rounding errors, which for the moment are ignored.

Theorems 6.2 and 6.3 show that $X_0 = e^{A+\Delta A}$ and $L_0 = L_{\exp}(A + \Delta A, E + \Delta E)$ with the *same* $\Delta A = 2^s g_m(2^{-s}A)$. We already know from the analysis in [5] how to choose s and m to keep ΔA acceptably small. It remains to investigate the norm of $\Delta E = L_{g_m}(2^{-s}A, E)$.

Let $\widetilde{g}_m(x) = \sum_{k=2m+1}^{\infty} c_k x^k$ be the power series resulting from replacing the coefficients of the power series expansion of the function $g_m(x) = \log(e^{-x} r_m(x))$ by their absolute values. Using the second bound in (3.5) we have

$$\frac{\|\Delta E\|}{\|E\|} = \frac{\|L_{g_m}(2^{-s}A, E)\|}{\|E\|} \le \|L_{g_m}(2^{-s}A)\| \le \widetilde{g}_m{}'(\theta), \tag{6.10}$$

where $\theta = \|2^{-s}A\|$. Define $\ell_m = \max\{\, \theta : \widetilde{g}_m{}'(\theta) \le u \,\}$, where $u = 2^{-53} \approx 1.1 \times 10^{-16}$ is the unit roundoff for IEEE double precision arithmetic. Using MATLAB's Symbolic Math Toolbox we evaluated ℓ_m, $m = 1\colon 20$, by summing the first 150 terms of the series symbolically in 250 decimal digit arithmetic. Table 6.1 shows these values along with analogous values θ_m calculated in [5], which are the maximal values of θ such that a bound on $\|\Delta A\|/\|A\|$ obtained from (6.7) does not exceed u. In every case $\ell_m < \theta_m$, which is not surprising given that we are approximating L_{r_m} by an approximation chosen for computational convenience rather than its approximation properties, but

the ratio θ_m/ℓ_m is close to 1. For each m, if $\theta \le \ell_m$ then we are assured that

$$X_0 = e^{A+\Delta A}, \quad L_0 = L_{\exp}(A+\Delta A, E+\Delta E), \quad \|\Delta A\| \le u\|A\|, \quad \|\Delta E\| \le u\|E\|;$$

in other words, perfect backward stability is guaranteed for such θ.

In order to develop an algorithm we now need to look at the cost of evaluating $r_m = p_m/q_m$ and L_{r_m}, where r_m is the $[m/m]$ Padé approximant to e^x. Higham [5] shows how to efficiently evaluate $p_m(A)$ and $q_m(A)$ by using one type of scheme for $m \le 11$ and another for $m \ge 12$; the number of matrix multiplications, π_m, required to compute $p_m(A)$ and $q_m(A)$ is given in [5, Table 2.2]. As Theorem 4.1 suggests, the Fréchet derivatives L_{p_m} and L_{q_m} can be calculated at an extra cost of $2\pi_m$ multiplications by differentiating the schemes for p_m and q_m. We now give the details.

We consider the odd degree Padé approximants to the exponential function. Analogous techniques apply to the even degree approximants (which, as in Algorithm 6.1, it will turn out we do not need). For $m = 3, 5, 7, 9$, we decompose $p_m = \sum_{i=0}^m b_i x^i$ into its odd and even parts:

$$p_m(x) = x \sum_{k=0}^{(m-1)/2} b_{2k+1} x^{2k} + \sum_{k=0}^{(m-1)/2} b_{2k} x^{2k} =: u_m(x) + v_m(x). \tag{6.11}$$

It follows that $q_m(x) = -u_m(x) + v_m(x)$ since $q_m(x) = p_m(-x)$, and hence

$$L_{p_m} = L_{u_m} + L_{v_m}, \qquad L_{q_m} = -L_{u_m} + L_{v_m}.$$

We obtain $L_{u_m}(A,E)$ and $L_{v_m}(A,E)$ by differentiating u_m and v_m, respectively:

$$L_{u_m}(A,E) = A \sum_{k=1}^{(m-1)/2} b_{2k+1} M_{2k} + E \sum_{k=0}^{(m-1)/2} b_{2k+1} A^{2k} \tag{6.12}$$

$$L_{v_m}(A,E) = \sum_{k=1}^{(m-1)/2} b_{2k} M_{2k}. \tag{6.13}$$

The $M_k = L_{x^k}(A,E)$ are computed using (3.3).

For $m = 13$ it is more efficient to use the odd-even splitting $p_{13} = u_{13} + v_{13}$, where

$$\begin{aligned} u_{13}(x) &= xw(x), \quad w(x) = x^6 w_1(x) + w_2(x), \quad v_{13}(x) = x^6 z_1(x) + z_2(x), \\ w_1(x) &= b_{13}x^6 + b_{11}x^4 + b_9 x^2, \quad w_2(x) = b_7 x^6 + b_5 x^4 + b_3 x^2 + b_1, \\ z_1(x) &= b_{12}x^6 + b_{10}x^4 + b_8 x^2, \quad z_2(x) = b_6 x^6 + b_4 x^4 + b_2 x^2 + b_0. \end{aligned}$$

Differentiating these polynomials yields

$$\begin{aligned} L_{u_{13}}(A,E) &= AL_w(A,E) + Ew(A), \\ L_{v_{13}}(A,E) &= A^6 L_{z_1}(A,E) + M_6 z_1(A) + L_{z_2}(A,E), \end{aligned}$$

where

$$\begin{aligned} L_w(A,E) &= A^6 L_{w_1}(A,E) + M_6 w_1(A) + L_{w_2}(A,E), \\ L_{w_1}(A,E) &= b_{13}M_6 + b_{11}M_4 + b_9 M_2, \\ L_{w_2}(A,E) &= b_7 M_6 + b_5 M_4 + b_3 M_2, \\ L_{z_1}(A,E) &= b_{12}M_6 + b_{10}M_4 + b_8 M_2, \\ L_{z_2}(A,E) &= b_6 M_6 + b_4 M_4 + b_2 M_2. \end{aligned}$$

TABLE 6.2
Number of matrix multiplications, ω_m, required to evaluate $r_m(A)$ and $L_{r_m}(A,E)$, and measure of overall cost C_m in (6.17).

m	1	2	3	4	5	6	7	8	9	10
ω_m	1	4	7	10	10	13	13	16	16	19
C_m	25.5	12.5	8.5	6.9	5.3	5.2	4.4	4.7	4.2	4.7

m	11	12	13	14	15	16	17	18	19	20
ω_m	19	19	19	22	22	22	22	25	25	25
C_m	4.4	4.0	3.8	4.5	4.3	4.1	3.9	4.7	4.6	4.5

Then $L_{p_{13}} = L_{u_{13}} + L_{v_{13}}$ and $L_{q_{13}} = -L_{u_{13}} + L_{v_{13}}$. We finally solve for $r_m(A)$ and $L_{r_m}(A,E)$ the equations

$$(-u_m + v_m)(A)r_m(A) = (u_m + v_m)(A), \tag{6.14}$$

$$(-u_m + v_m)(A)L_{r_m}(A,E) = (L_{u_m} + L_{v_m})(A,E) + (L_{u_m} - L_{v_m})(A,E)r_m(A). \tag{6.15}$$

We are now in a position to choose the degree m and the scaling parameter s. Table 6.2 reports the total number of matrix multiplications, $\omega_m = 3\pi_m + 1$, necessary to evaluate r_m and L_{r_m} for a range of m, based on [5, Table 2.2] and the observations above. In evaluating the overall cost we need to take into account the squaring phase. If $\|A\| > \ell_m$ then in order to use the $[m/m]$ Padé approximant we must scale A by 2^{-s} so that $\|2^{-s}A\| \le \ell_m$, that is, we need $s = \lceil \log_2(\|A\|/\ell_m) \rceil$. From the recurrence (6.4), we see that $3s$ matrix multiplications are added to the cost of evaluating r_m and L_{r_m}. Thus the overall cost in matrix multiplications is

$$\omega_m + 3s = 3\pi_m + 1 + 3\max(\lceil \log_2 \|A\| - \log_2 \ell_m \rceil, 0). \tag{6.16}$$

To minimize the cost we therefore choose m to minimize the quantity

$$C_m = \pi_m - \log_2 \ell_m, \tag{6.17}$$

where we have dropped the constant terms and factors in (6.16). Table 6.2 reports the C_m values. The table shows that $m = 13$ is the optimal choice, just as it is for the scaling and squaring method for the exponential itself [5]. The ω_m values also show that only $m = 1, 2, 3, 5, 7, 9$ need be considered if $\|A\| < \ell_{13}$. As in [5] we rule out $m = 1$ and $m = 2$ on the grounds of possible loss of significant figures in floating point arithmetic.

It remains to check that the evaluation of L_{p_m}, L_{q_m}, and L_{r_m} is done accurately in floating point arithmetic. The latter matrix is evaluated from (6.15), which involves solving a matrix equation with coefficient matrix $q_m(A)$, just as in the evaluation of r_m, and the analysis from [5] guarantees that $q_m(A)$ is well conditioned for the scaled A. It can be shown that for our schemes for evaluating L_{p_m} we have

$$\|L_{p_m}(A,E) - fl(L_{p_m}(A,E))\|_1 \le \widetilde{\gamma}_{n^2} p'_m(\|A\|_1)\|E\|_1 \approx \widetilde{\gamma}_{n^2} e^{\|A\|_1/2}\|E\|_1,$$

where we have used the facts that p_m has positive coefficients and $p_m(x) \approx e^{x/2}$. Here, $\widetilde{\gamma}_k = cku/(1-cku)$, where c denotes a small integer constant. At least in an absolute sense, this bound is acceptable for $\|A\| \le \ell_{13}$. An entirely analogous bound can be obtained for L_{q_m}, since $q_m(x) = p_m(-x)$.

We now state the complete algorithm.

ALGORITHM 6.4 (scaling and squaring algorithm for exponential and Fréchet derivative). *Given $A, E \in \mathbb{C}^{n\times n}$ this algorithm computes $R = e^A$ and $L = L_{\exp}(A, E)$ by a scaling and squaring algorithm. It uses the parameters ℓ_m listed in Table* 6.1. *The algorithm is intended for IEEE double precision arithmetic.*

```
1  for m = [3 5 7 9]
2      if ‖A‖_1 ≤ ℓ_m
3         Evaluate U = u_m(A) and V = v_m(A), using (6.11).
4         Evaluate L_u = L_{u_m}(A, E) and L_v = L_{v_m}(A, E), using (6.12) and (6.13).
5         s = 0; goto line 26
6      end
7  end
8  s = ⌈log_2(‖A‖_1/ℓ_13)⌉, the minimal integer such that ‖2^{-s}A‖_1 ≤ ℓ_13.
9  A ← 2^{-s}A and E ← 2^{-s}E
10 A_2 = A^2, A_4 = A_2^2, A_6 = A_2A_4
11 M_2 = AE + EA, M_4 = A_2M_2 + M_2A_2, M_6 = A_4M_2 + M_4A_2
12 W_1 = b_13A_6 + b_11A_4 + b_9A_2
13 W_2 = b_7A_6 + b_5A_4 + b_3A_2 + b_1I
14 Z_1 = b_12A_6 + b_10A_4 + b_8A_2
15 Z_2 = b_6A_6 + b_4A_4 + b_2A_2 + b_0I
16 W = A_6W_1 + W_2
17 U = AW
18 V = A_6Z_1 + Z_2
19 L_{w_1} = b_13M_6 + b_11M_4 + b_9M_2
20 L_{w_2} = b_7M_6 + b_5M_4 + b_3M_2
21 L_{z_1} = b_12M_6 + b_10M_4 + b_8M_2
22 L_{z_2} = b_6M_6 + b_4M_4 + b_2M_2
23 L_w = A_6L_{w_1} + M_6W_1 + L_{w_2}
24 L_u = AL_w + EW
25 L_v = A_6L_{z_1} + M_6Z_1 + L_{z_2}
26 Solve (−U + V)R = U + V for R.
27 Solve (−U + V)L = L_u + L_v + (L_u − L_v)R for L.
28 for k = 1: s
29     L ← RL + LR
30     R ← R^2
31 end
```

Cost: $(\omega_m + 3s)M + 2D$, where m is the degree of Padé approximant used and ω_m is given in Table 6.2. The linear systems at lines 26 and 27 have the same coefficient matrix, so an LU factorization can computed once and reused.

Since $L_{\exp}(A, \alpha E) = \alpha L_{\exp}(A, E)$, an algorithm for computing $L_{\exp}(A, E)$ should not be influenced in any significant way by $\|E\|$, and this is the case for Algorithm 6.4. Najfeld and Havel [15] propose computing $L_{\exp}(A, E)$ using their version of the scaling and squaring method for the exponential in conjunction with (2.1). With this approach E affects the amount of scaling, and overscaling results when $\|E\| \gg \|A\|$, while how to scale E to produce the most accurate result is unclear.

To assess the cost of Algorithm 6.4 we compare it with Algorithm 6.1 and with a "Kronecker–Sylvester scaling and squaring algorithm" of Kenney and Laub [11], which is based on a Kronecker sum representation of the Fréchet derivative. In the form detailed in [6, section 10.6.2], this latter algorithm scales to obtain $\|2^{-t}A\| \le 1$, evaluates the [8/8] Padé approximant to $\tanh(x)/x$ at the scaled Kronecker sum, and then uses the recurrence (6.4) or the variant (6.3) that explicitly computes $X_i = e^{2^{-i}A}$

in each step. It requires one matrix exponential, $(17 + 3t)M$, and the solution of 8 Sylvester equations if (6.4) is used, or s matrix exponentials, $(18 + 2t)M$, and the same number of Sylvester equation solutions if (6.3) is used.

To compare the algorithms, assume that the Padé degree $m = 13$ is used in Algorithms 6.1 and 6.4. Then Algorithm 6.4 costs $(19+3s)M+2D$ and Algorithm 6.1 costs $(6+s)M+D$. Two conclusions can be drawn. First, Algorithm 6.4 costs about three times as much as just computing e^A. Second, since the cost of solving a Sylvester equation is about $60n^3$ flops, which is the cost of 30 matrix multiplications, the Kronecker–Sylvester algorithm is an order of magnitude more expensive than Algorithm 6.4. To be more specific, consider the case where $\|A\| = 9$, so that $s = 1$ in Algorithms 6.1 and 6.4 and $t = 4$, and ignore the cost of computing the matrix exponential in the less expensive "squaring" variant of the Kronecker–Sylvester algorithm. Then the operation counts in flops are approximately $48n^3$ for Algorithm 6.4 (e^A and $L_{\exp}(A, E)$), $16n^3$ for Algorithm 6.1 (e^A only), and $538n^3$ for the Kronecker–Sylvester algorithm ($L_{\exp}(A, E)$ only). A further drawback of the Kronecker–Sylvester algorithm is that it requires complex arithmetic, so the effective flop count is even higher.

Other algorithms for $L_{\exp}(A, E)$ are those of Kenney and Laub [9] and Mathias [12] (see also [6, section 10.6.1]), which apply quadrature to an integral representation of the Fréchet derivative. These algorithms are intended only for low accuracy approximations and do not lend themselves to combination with Algorithm 6.1.

We describe a numerical experiment, modeled on that in [5], that tests the accuracy of Algorithm 6.4. We took 74 test matrices, which include some from MATLAB (in particular, from the `gallery` function), some from the Matrix Computation Toolbox [3], and test matrices from the e^A literature; most matrices are 10×10, with a few having smaller dimension. We evaluated the normwise relative errors of the computed Fréchet derivatives $L_{\exp}(A, E)$, using a different E, generated as `randn(n)`, for each A. The "exact" Fréchet derivative is obtained using (2.1) with the exponential evaluated at 100 digit precision via MATLAB's Symbolic Math Toolbox. Figure 6.1 displays

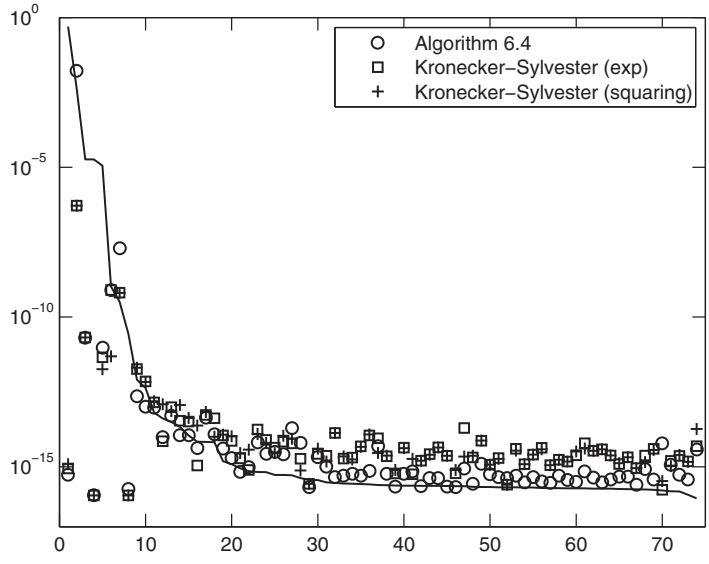


FIG. 6.1. *Normwise relative errors in Fréchet derivatives* $L_{\exp}(A, E)$ *computed by Algorithm* 6.4 *and two variants of the Kronecker–Sylvester algorithm for* 74 *matrices* A *with a different random* E *for each* A*, along with estimate of* $\mathrm{cond}(L_{\exp}, A)u$ (*solid line*).

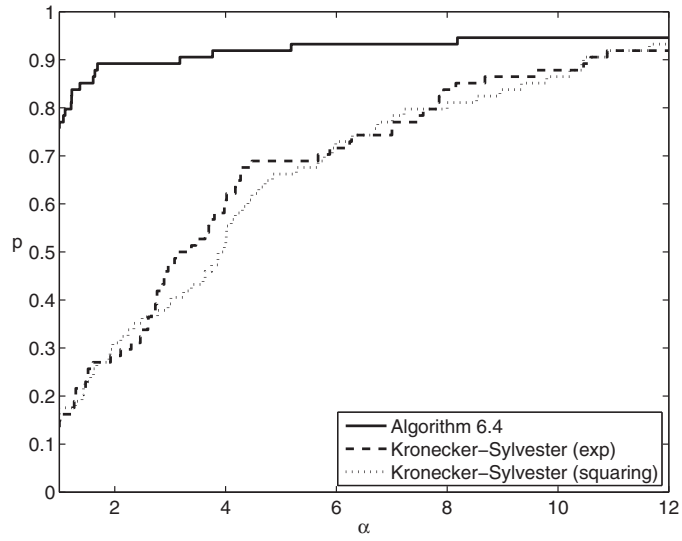


FIG. 6.2. *Same data as in Figure* 6.1 *presented as a performance profile.*

the Frobenius norm relative errors for Algorithm 6.4 and for the Kronecker–Sylvester algorithm in both "squaring" and "exponential" variants. Also shown is a solid line representing a finite difference approximation to $\mathrm{cond}(L_{\exp}, A)u$, where $\mathrm{cond}(L_{\exp}, A)$ is a condition number defined in terms of the Jacobian of the map L regarded as a function of A and E (we use (1.2) with a small, random E); this line indicates the accuracy we would expect from a forward stable algorithm for computing the Fréchet derivative. Figure 6.1 shows that all the methods are performing in a reasonably forward stable manner but does not clearly reveal differences between the methods.

Figure 6.2 plots the same data as a performance profile: for a given α the corresponding point on each curve indicates the fraction p of problems on which the method had error at most a factor α times that of the smallest error over all three methods. The results show clear superiority of Algorithm 6.4 over the Kronecker–Sylvester algorithm in terms of accuracy, for both variants of the latter algorithm. Since Algorithm 6.4 is also by far the more efficient, as explained above, it is clearly the preferred method.

7. Condition number estimation. We now turn our attention to estimating the condition number of the matrix exponential, which from (1.2) is

$$\kappa_{\exp}(A) = \frac{\|L_{\exp}(A)\|\,\|A\|}{\|e^A\|}.$$

Algorithm 6.4 can compute $L_{\exp}(A, E)$ for any direction E, but to obtain the norm $\|L_{\exp}(A)\|$ we need to maximize $L_{\exp}(A, E)$ over all E of unit norm.

For the moment we will consider general f. We can write

$$\mathrm{vec}(L(A, E)) = K(A)\mathrm{vec}(E), \tag{7.1}$$

where $K(A) \in \mathbb{C}^{n^2 \times n^2}$ is independent of E and $\mathrm{vec}(E) \in \mathbb{C}^{n^2}$ denotes the vector comprising the columns of E strung one on top of the other from first to last. We refer to $K(A)$ as the Kronecker form of the Fréchet derivative. From (7.1) we have

$\|L(A,E)\|_F = \|K(A)\mathrm{vec}(E)\|_2$, and on dividing by $\|E\|_F = \|\mathrm{vec}(E)\|_2$ and maximizing over all E it follows that

$$\|L(A)\|_F = \|K(A)\|_2. \tag{7.2}$$

Therefore we can compute $\|L(A)\|_F$ exactly by forming $K(A)$, whose columns are $\mathrm{vec}(L(A, e_i e_j^T))$ for $i,j = 1\colon n$, and then taking the 2-norm. This is a prohibitively expensive computation, typically requiring $O(n^5)$ flops. However, in practice only an estimate of the correct order of magnitude is needed. For this purpose it is appropriate to use matrix norm estimation techniques.

The following algorithm is essentially the usual power method applied to $K(A)$, and exploits the relation (7.2) [6, section 3.4], [9].

ALGORITHM 7.1 (power method on Fréchet derivative). *Given $A \in \mathbb{C}^{n\times n}$ and the Fréchet derivative L of a function f, this algorithm uses the power method to produce an estimate $\eta \le \|L(A)\|_F$.*

```
1  Choose a nonzero starting matrix Z_0 ∈ C^{n×n}.
2  for k = 0: ∞
3        W_{k+1} = L(A, Z_k)
4        Z_{k+1} = L*(A, W_{k+1})
5        η_{k+1} = ||Z_{k+1}||_F / ||W_{k+1}||_F
6        if converged, η = η_{k+1}, quit, end
7  end
```

Here, $\star$ denotes the adjoint, and for the exponential, $L^{\star}_{\exp}(A,W) \equiv L_{\exp}(A^*, W)$.

We do not specify Algorithm 7.1 in any more detail because we prefer a 1-norm variant of the power method. For the 1-norm there is no analogue of (7.2), but the next lemma shows how $\|K(A)\|_1$ compares with $\|L(A)\|_1$.

LEMMA 7.2 ([6, Lemma 3.18]). *For $A \in \mathbb{C}^{n\times n}$ and any function f,*

$$\frac{\|L(A)\|_1}{n} \le \|K(A)\|_1 \le n\|L(A)\|_1. \qquad \square \tag{7.3}$$

The following algorithm, which again needs the ability to evaluate $L(A,E)$ and $L^{\star}(A,E)$, is essentially [6, Algorithm 3.22]; it employs a block 1-norm estimation algorithm of Higham and Tisseur [7], which for an $n \times n$ matrix carries out a 1-norm power iteration whose iterates are $n \times t$ matrices, where t is a parameter.

ALGORITHM 7.3 (block 1-norm estimator for Fréchet derivative). *Given a matrix $A \in \mathbb{C}^{n\times n}$ this algorithm uses a block 1-norm estimator to produce an estimate η of $\|L(A)\|_1$. More precisely, $\eta \le \|K(A)\|_1$, where $\|K(A)\|_1$ satisfies* (7.3).

1 Apply Algorithm 2.4 from Higham and Tisseur [7] with parameter $t = 2$ to the Kronecker matrix representation $B := K(A)$ of $L(A)$, noting that $By \equiv \mathrm{vec}(L(A,E))$ and $B^*y \equiv \mathrm{vec}(L^{\star}(A,E))$, where $\mathrm{vec}(E) = y$.

Key properties of Algorithm 7.3 are that it typically requires about $4t$ Fréchet derivative evaluations and it almost invariably produces an estimate of $\|K(A)\|_1$ correct to within a factor 3. A factor n of uncertainty is added when we take η as an estimate of $\|L(A)\|_1$, but just changing the norm from the 1-norm to the ∞-norm can introduce such a factor, so it is not a serious weakness. Overall, the algorithm is a very reliable means of estimating $\|L(A)\|_1$ to within a factor $3n$.

Returning to the exponential, our interest is in how to combine Algorithms 6.4 and 7.3 in the most efficient manner. We need to evaluate $L(A,E)$ and $L(A^*,E)$ for a fixed A and several different E, without knowing all the E at the start of the

TABLE 7.1
Matrices that must be computed and stored during the initial e^A evaluation, to be reused during the Fréchet derivative evaluations. "LU fact" stands for LU factorization of $-u_m + v_m$, and $B = A/2^s$.

m						
3	$r_3(A)$				LU fact.	$W_3(A)$
5	$r_5(A)$	A^2			LU fact.	$W_5(A)$
7	$r_7(A)$	A^2	A^4		LU fact.	$W_7(A)$
9	$r_9(A)$	A^2	A^4		LU fact.	$W_9(A)$
13	$r_{13}(B)^{2^i}$, $i = 0\colon s-1$	B^2	B^4	B^6	LU fact.	W

computation. To do so we will store matrices accrued during the initial computation of e^A and reuse them in the Fréchet derivative evaluations. This of course assumes the availability of extra storage, but in modern computing environments ample storage is usually available.

In view of the evaluation schemes (6.11)–(6.13) and (6.15), for $m \in \{3, 5, 7, 9\}$ we need to store A^{2k}, $k = 1\colon d_{(m-1)/2}$, where $d = [0\ 1\ 2\ 2]$, along with $W_m(A) = \sum_{k=0}^{(m-1)/2} b_{2k+1}A^{2k}$, $r_m(A)$, and the LU factorization of $(-u_m + v_m)(A)$. For $m = 13$, the matrix A needs to be scaled, to $B = A/2^s$. According to the scheme used in Algorithm 6.4 we need to store B^{2k}, $k = 1\colon 3$, $W \equiv w(B)$, the LU factorization of $(-u_m + v_m)(B)$, and $r_m(B)^{2^i}$, $i = 0\colon s-1$. Table 7.1 summarizes the matrices that need to be stored for each m.

The following algorithm computes the matrix exponential and estimates its condition number. Since the condition number is not needed to high accuracy we use the parameters θ_m in Table 6.1 (designed for e^A) instead of ℓ_m (designed for $L(A, E)$). The bound in (6.10) for the Fréchet derivative backward error $\|\Delta E\|/\|E\|$ does not exceed $28u$ for $m \le 13$ when we use the θ_m, so the loss in backward stability for the Fréchet derivative evaluation is negligible. If the condition estimate is omitted, the algorithm reduces to Algorithm 6.1. The algorithm exploits the relation $L_f(A^*, E) = L_f(A, E^*)^*$, which holds for any f with a power series expansion with real coefficients, by (3.1).

ALGORITHM 7.4 (scaling and squaring algorithm for the matrix exponential with 1-norm condition estimation). *Given $A \in \mathbb{C}^{n\times n}$ this algorithm computes $X = e^A$ by the scaling and squaring method (Algorithm 6.1) and an estimate $\gamma \approx \kappa_{\exp}(A)$ using the block 1-norm estimator (Algorithm 7.1). It uses the values θ_m listed in Table 6.1. The algorithm is intended for IEEE double precision arithmetic.*

```
1  α = ‖A‖_1
2  for m = [3 5 7 9]
3      if α ≤ θ_m
4          Evaluate U = u_m(A) and V = v_m(A), using (6.11).
5          Solve (−U + V)X = U + V for X.
6          Store the needed matrices (see Table 7.1).
7          Use Algorithm 7.3 to produce an estimate η ≈ ‖L_exp(A)‖_1.
           ..... To compute L_exp(A, E) for a given E:
8                  Evaluate M_2k = L_{x^{2^k}}(A, E), k = 1 : (m − 1)/2.
9                  L_u ← A (Σ_{k=1}^{(m−1)/2} b_{2k+1} M_2k) + E W_m(A)
10                 L_v ← Σ_{k=1}^{(m−1)/2} b_2k M_2k
11                 Solve (−U + V)L = L_u + L_v + (L_u − L_v)X for L.
           ..... To compute L*_exp(A, E) for a given E:
```

12 Execute lines 8–11 with E replaced by its conjugate transpose and take the conjugate transpose of the result.
13 goto line 44
14 end
15 % Use degree $m = 13$.
16 $s = \lceil \log_2(\alpha/\theta_{13}) \rceil$, the minimal integer such that $2^{-s}\alpha \le \theta_{13}$.
17 $A \leftarrow 2^{-s}A$
18 $A_2 = A^2$, $A_4 = A_2^2$, $A_6 = A_2A_4$
19 $W_1 = b_{13}A_6 + b_{11}A_4 + b_9A_2$
20 $Z_1 = b_{12}A_6 + b_{10}A_4 + b_8A_2$
21 $W = A_6W_1 + b_7A_6 + b_5A_4 + b_3A_2 + b_1I$
22 $U = AW$
23 $V = A_6Z_1 + b_6A_6 + b_4A_4 + b_2A_2 + b_0I$
24 Solve $(-U + V)X_s = U + V$ for X_s
25 for $i = s\colon -1\colon 1$
26 $X_{i-1} = X_i^2$
27 end
28 $X = X_0$
29 Use Algorithm 7.3 to produce an estimate $\eta \approx \|L_{\exp}(\widetilde{A})\|_1$, where $\widetilde{A}$ denotes the original input matrix A.
..... To compute $L_{\exp}(\widetilde{A}, E)$ for a given E:
30 $E \leftarrow 2^{-s}E$
31 $M_2 = AE + EA$, $M_4 = A_2M_2 + M_2A_2$, $M_6 = A_4M_2 + M_4A_2$
32 $L_{w_1} = b_{13}M_6 + b_{11}M_4 + b_9M_2$
33 $L_{w_2} = b_7M_6 + b_5M_4 + b_3M_2$
34 $L_{z_1} = b_{12}M_6 + b_{10}M_4 + b_8M_2$
35 $L_{z_2} = b_6M_6 + b_4M_4 + b_2M_2$
36 $L_w = A_6L_{w_1} + M_6W_1 + L_{w_2}$
37 $L_u = AL_w + EW$
38 $L_v = A_6L_{z_1} + M_6Z_1 + L_{z_2}$
39 Solve $(-U + V)L = L_u + L_v + (L_u - L_v)X_s$ for L.
40 for $i = s\colon -1\colon 1$
41 $L \leftarrow X_iL + LX_i$
42 end
..... To compute $L^\star_{\exp}(\widetilde{A}, E)$ for a given E:
43 Execute lines 30–42 with E replaced by its conjugate transpose and take the conjugate transpose of the result.
44 $\gamma = \eta\alpha/\|X\|_1$

The cost of Algorithm 7.4 is the cost of computing e^A plus the cost of about 8 Fréchet derivative evaluations, so obtaining e^A and the condition estimate multiplies the cost of obtaining just e^A by a factor of about 17. This factor can be reduced to 9 if the parameter t in the block 1-norm power method is reduced to 1, at a cost of slightly reduced reliability.

In our MATLAB implementation of Algorithm 7.4 we invoke the function `funm_condest1` from the Matrix Function Toolbox [4], which interfaces to the MATLAB function `normest1` that implements the block 1-norm estimation algorithm of [7].

With the same matrices as in the test of the previous section we used Algorithm 7.4 to estimate $\|K(A)\|_1$ and also computed $\|K(A)\|_1$ exactly by forming $K(A)$ as described at the start of this section. Figure 7.1 plots the norms and the estimates.

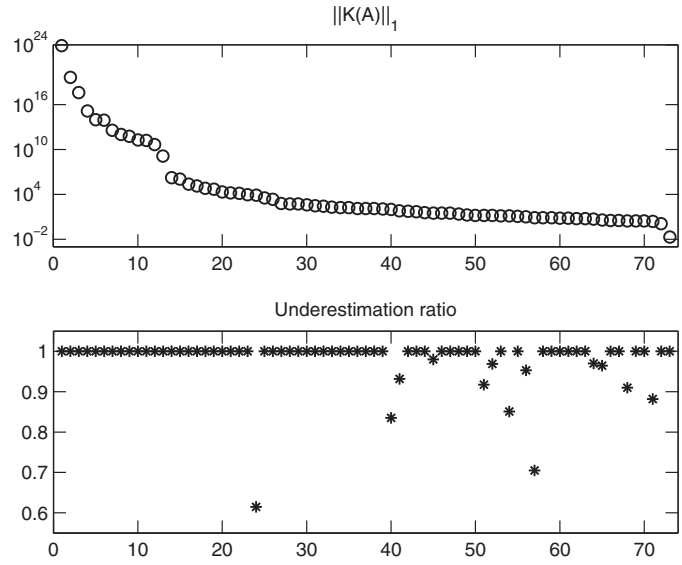


FIG. 7.1. $\|K(A)\|_1$ *and underestimation ratio* $\eta/\|K(A)\|_1$*, where* η *is the estimate of* $\|K(A)\|_1$ *produced by Algorithm* 7.4.

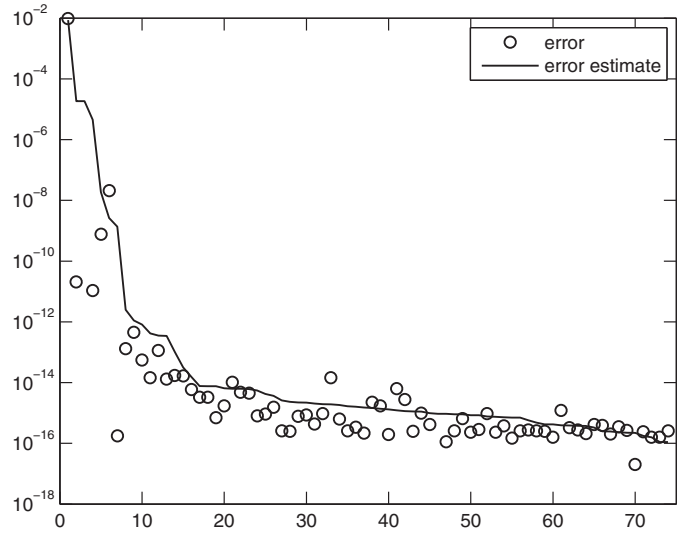


FIG. 7.2. *Normwise relative error for computed exponential and error estimate comprising condition number estimate times unit roundoff.*

The worst underestimation ratio is 0.61, so the estimates are all within a factor 2 of the true 1-norm.

Finally, we invoked Algorithm 7.4 on the same set of test matrices and computed the "exact" exponential in 100 digit precision. Figure 7.2 plots the error in the computed exponential along with the quantity γu: the condition estimate multiplied by the unit roundoff, regarded as an error estimate. If the scaling and squaring algorithm were forward stable and the condition estimate reliable we would expect

the error to be bounded by $\phi(n)\gamma u$ for some low degree polynomial ϕ. The overall numerical stability of the scaling and squaring algorithm is not understood [6], but our experience is that the method usually does behave in a forward stable way. Figure 7.2 indicates that the condition estimate from Algorithm 7.4 provides a useful guide to the accuracy of the computed exponential from the algorithm.

8. Concluding remarks. The LAPACK Users' Guide states [1, p. 77] that "Our goal is to provide error bounds for most quantities computed by LAPACK." This is a desirable goal for any numerical algorithm, and in order to achieve it error analysis must be developed that yields a reasonably sharp error bound that can be efficiently estimated. For matrix function algorithms a complete error analysis is not always available, and for the forward error a bound of the form $\mathrm{cond}(f, A)u$ is the best we can expect in general. To date relatively little attention has been paid to combining evaluation of $f(A)$ with computation of the Fréchet derivative $L(A, E)$ and estimation of the condition number $\mathrm{cond}(f, A)$. We are currently applying and extending the ideas developed here to other transcendental functions such as the logarithm and the sine and cosine and will report on this work in a future paper.

Acknowledgment. We thank Bruno Iannazzo for his helpful comments on a draft of this paper.

REFERENCES

[1] E. Anderson, Z. Bai, C. H. Bischof, S. Blackford, J. W. Demmel, J. J. Dongarra, J. J. Du Croz, A. Greenbaum, S. J. Hammarling, A. McKenney, and D. C. Sorensen, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, PA, 1999.

[2] G. A. Baker, Jr., and P. Graves-Morris, *Padé Approximants*, Encyclopedia of Mathematics and Its Applications, 2nd ed., Vol. 59, Cambridge University Press, Cambridge, 1996.

[3] N. J. Higham, *The Matrix Computation Toolbox*, http://www.ma.man.ac.uk/~higham/mctoolbox.

[4] N. J. Higham, *The Matrix Function Toolbox*, http://www.ma.man.ac.uk/~higham/mftoolbox.

[5] N. J. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1179–1193.

[6] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, PA, 2008.

[7] N. J. Higham and F. Tisseur, *A block algorithm for matrix* 1*-norm estimation, with an application to* 1*-norm pseudospectra*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1185–1201.

[8] G. Jewitt and J. R. McCrorie, *Computing estimates of continuous time macroeconometric models on the basis of discrete data*, Comput. Statist. Data Anal., 49 (2005), pp. 397–416.

[9] C. S. Kenney and A. J. Laub, *Condition estimates for matrix functions*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 191–209.

[10] C. S. Kenney and A. J. Laub, *Polar decomposition and matrix sign function condition estimates*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 488–504.

[11] C. S. Kenney and A. J. Laub, *A Schur–Fréchet algorithm for computing the logarithm and exponential of a matrix*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 640–663.

[12] R. Mathias, *Evaluating the Frechet derivative of the matrix exponential*, Numer. Math., 63 (1992), pp. 213–226.

[13] R. Mathias, *A chain rule for matrix functions and applications*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 610–620.

[14] C. B. Moler and C. F. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Rev., 45 (2003), pp. 3–49.

[15] I. Najfeld and T. F. Havel, *Derivatives of the matrix exponential and their computation*, Adv. Appl. Math., 16 (1995), pp. 321–375.

[16] M. S. Paterson and L. J. Stockmeyer, *On the number of nonscalar multiplications necessary to evaluate polynomials*, SIAM J. Comput., 2 (1973), pp. 60–66.

[17] R. C. Ward, *Numerical computation of the matrix exponential with accuracy estimate*, SIAM J. Numer. Anal., 14 (1977), pp. 600–610.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1658–1676

FAST CONDITION ESTIMATION FOR A CLASS OF STRUCTURED EIGENVALUE PROBLEMS*

A. J. LAUB† AND J. XIA‡

Abstract. We present a fast condition estimation algorithm for the eigenvalues of a class of structured matrices. These matrices are low rank modifications to Hermitian, skew-Hermitian, and unitary matrices. Fast structured operations for these matrices are presented, including Schur decomposition, eigenvalue block swapping, matrix equation solving, compact structure reconstruction, etc. Compact semiseparable representations of matrices are used in these operations. We use these operations in a new, improved version of the statistical condition estimation method for eigenvalue problems. The estimation algorithm costs $O(n^2)$ flops for all eigenvalues, instead of $O(n^3)$ as in traditional algorithms, where n is the order of the matrix. The algorithm provides reliable condition estimates for both eigenvalues and eigenvalue clusters. The proposed structured matrix operations are also useful for additional eigenvalue problems and other applications. Numerical examples are used to illustrate the reliability and efficiency of the algorithm.

Key words. statistical condition estimation, low rank modification, sequentially semiseparable structure, structured Schur decomposition, diagonal block swapping, structured Sylvester equation

AMS subject classifications. 65F15, 65F30, 65F35

DOI. 10.1137/070707713

1. Introduction.

The condition number of an eigenvalue or eigenvalue cluster measures the sensitivity of the eigenvalue or eigenvalue cluster to small changes in the input matrix, and it may be used to bound the error in the computed approximation. For a general order-n matrix, it usually costs $O(n^3)$ flops or more to estimate the sensitivity of all its eigenvalues, for example, by considering separations of eigenvalues, considering angles between left eigenvectors and right eigenvectors, and other methods [1], [6], [19], [21], [28]. For structured eigenvalue problems, on the one hand, it is important to capture the structures [13]. We can take advantage of the structures to obtain fast condition estimation. The development of new fast structured algorithms for these matrices in recent years makes it possible to obtain fast condition estimation. In this paper, we consider condition estimation for the eigenvalues of a class of structured matrices, and we present a reliable estimation scheme which costs only $O(n^2)$ flops. This class of matrices has the following structures:

1. Low rank modifications to Hermitian and skew-Hermitian matrices. Examples include Frobenius matrices, diagonal plus rank one matrices, and arrowhead matrices which arise in applications such as bidiagonal SVD, divide-and-conquer algorithms for some eigenvalue problems [18], etc.
2. Low rank modifications to unitary matrices such as companion matrices, which are closely related to the problems of finding polynomial roots and solving certain differential equations.

Any such matrix $C \in \mathbb{R}^{n\times n}$ is a low rank perturbation to a *rank symmetric* matrix [3] (a matrix $\hat{C}$ is said to be rank symmetric if for any 2×2 block partition of

*Received by the editors November 9, 2007; accepted for publication (in revised form) by M. Van Barel September 9, 2008; published electronically January 23, 2009.
http://www.siam.org/journals/simax/30-4/70771.html

†Department of Electrical Engineering and Department of Mathematics, University of California, Los Angeles, CA 90095 (laub@ucla.edu).

‡Department of Mathematics, Purdue University, West Lafayette, IN 47907 (xiaj@math.purdue.edu).

$\hat{C} = \begin{bmatrix} \hat{C}_{11} & \hat{C}_{12} \\ \hat{C}_{21} & \hat{C}_{22} \end{bmatrix}$ with $\hat{C}_{11}$ and $\hat{C}_{22}$ square, the ranks of $\hat{C}_{12}$ and $\hat{C}_{21}$ are equal). That is,

$$C = \hat{C} + xy^T, \tag{1.1}$$

where $x, y \in \mathbb{C}^{n\times k}$ with $k \ll n$, and $\hat{C}$ is rank symmetric. Some fast methods for finding the eigenvalues of these matrices have been proposed in recent years (see, e.g., [3], [4], [5], [11], [16]). These methods exploit certain rank structure of the QR iterates when using QR iterations to find the eigenvalues. In this paper we show that the rank structure can also be used to accelerate the condition estimation of the eigenvalues.

As an example, the companion matrix

$$C = \begin{bmatrix} -a_{n-1} & -a_{n-2} & \cdots & -a_0 \\ 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & 0 \end{bmatrix} \tag{1.2}$$

can be written as

$$C = \begin{bmatrix} 0 & \cdots & 0 & \pm 1 \\ 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_0 \mp 1 \end{bmatrix}.$$

To quickly find the eigenvalues of (1.2), the fast structured QR iteration algorithm in [11] computes structured QR iterates which are unitarily similar to C. The iterations are done via the Q and R factors of the QR iterates. It can be shown that the QR iterates have small off-diagonal ranks [3], [5], [11]. Thus the Q and R factors can be efficiently represented by rank structures called *sequentially semiseparable* (SSS) matrix forms, proposed in [7], [8], [9]. An SSS matrix looks like

$$\begin{bmatrix} \mathcal{D}_1 & \mathcal{U}_1\mathcal{V}_2^T & \mathcal{U}_1\mathcal{W}_2\mathcal{V}_3^T & \mathcal{U}_1\mathcal{W}_2\mathcal{W}_3\mathcal{V}_4^T \\ \mathcal{P}_2\mathcal{Q}_1^T & \mathcal{D}_2 & \mathcal{U}_2\mathcal{V}_3^T & \mathcal{U}_2\mathcal{W}_3\mathcal{V}_4^T \\ \mathcal{P}_3\mathcal{R}_2\mathcal{Q}_1^T & \mathcal{P}_3\mathcal{Q}_2^T & \mathcal{D}_3 & \mathcal{U}_3\mathcal{V}_4^T \\ \mathcal{P}_4\mathcal{R}_3\mathcal{R}_2\mathcal{Q}_1^T & \mathcal{P}_4\mathcal{R}_3\mathcal{Q}_2^T & \mathcal{P}_4\mathcal{Q}_3^T & \mathcal{D}_4 \end{bmatrix}, \tag{1.3}$$

where the matrices $\{\mathcal{D}_i\}$, $\{\mathcal{U}_i\}$, $\{\mathcal{W}_i\}$, $\{\mathcal{V}_i\}$, $\{\mathcal{P}_i\}$, $\{\mathcal{R}_i\}$, $\{\mathcal{Q}_i\}$ are called (SSS) generators. SSS structures are useful for problems where the off-diagonal blocks have small ranks (see, e.g., [10], [11]). When the off-diagonal ranks of an SSS matrix are small and the sizes of $\{\mathcal{W}_i\}$ and $\{\mathcal{R}_i\}$ are close to the off-diagonal ranks, the matrix is said to be in *compact* SSS form. In such a situation, the matrix can be represented by only a linear amount of data, and operations on the compact SSS matrices are very efficient. For example, it costs only linear time to solve compact SSS linear systems and to multiply two compact SSS matrices with the same partition. More details can be found in [7], [8], [9], [10], [11]. The use of compact SSS matrices in the algorithm in [11] provides an $O(n^2)$ cost eigensolver for companion matrices (and polynomial root problems).

1.1. Main results. This paper shows that we can also exploit the rank structure of the above class of eigenvalue problems (1.1) to provide efficient condition estimation

for the eigenvalues. We use the *statistical condition estimation* (SCE) method by Kenney and Laub [23]. In [19], a perturbation analysis for the *average eigenvalues* of a general matrix based on SCE has been given, and an SCE condition estimator is provided. The cost is $O(n^3)$ flops for all eigenvalues. Here, we first improve the estimator in [19] from various points of view. Then we take into account the rank structure of the above class of matrices in SCE and extend the estimator in [19] to all the eigenvalues or eigenvalue clusters of these matrices. Given the facts that the related matrix computations become structured, and that SCE is good at respecting matrix structures, we can reduce the total condition estimation cost for all eigenvalues to $O(n^2)$.

We present the main idea based on the companion matrix (1.2). For convenience, we consider only real matrices and maintain real arithmetic in this paper. Similar techniques can be easily extended to other matrices in the above class. This is discussed in section 4. In this paper, the following structured matrix algorithms are used or developed:

1. Use the existing algorithm in [11] to compute a compact SSS Schur decomposition of C in $O(n^2)$ flops. Improve the algorithm so that the maximum generator size is smaller than the original one in [11] (see the next item).
2. Preserve *quasi-triangular* (a block upper triangular matrix with 1×1 and/or 2×2 diagonal blocks) Schur form when bringing any desired eigenvalue or eigenvalue cluster to any other position. This is done by swapping the contiguous 1×1 or 2×2 diagonal blocks of the Schur form in structured form. Quickly update the initial Schur form to recover a new compact Schur form. A recovery procedure in [11] is improved (with even less cost) so that it also works for quasi-triangular matrices, and the computational rank result is consistent with the theoretical prediction as well. The total cost for all eigenvalues is $O(n^2)$ flops.
3. Represent certain Sylvester equations in structured forms. Quickly solve a structured Sylvester equation for each eigenvalue so that the total cost for all eigenvalues is $O(n^2)$ instead of $O(n^3)$.
4. Use structured perturbation in SCE and evaluate all the condition estimates in $O(n^2)$ flops by taking advantage of matrix structures. Efficiently reconstruct certain invariant subspaces.

Our estimator works for both simple or multiple eigenvalues or eigenvalue clusters. The paper [26] presents some similar work. However, [26] requires that the eigenvalues of C all be distinct. The new operations here are more general, more efficient, and even simpler to implement than those in [26]. For example, after the diagonal block swapping, [26] uses SSS matrix multiplications to get the SSS representation of the new Schur form. But when the matrix has multiple eigenvalues, many SSS multiplications may be needed and the SSS Schur form may not be compact anymore. Instead, here we use a recovery strategy which always guarantees that the SSS Schur form is compact. As another example, here we simplify the reconstruction of the invariant subspace corresponding to an eigenvalue or eigencluster after the diagonal block swapping.

Similar techniques can also speed up the condition estimation for the eigenvectors of (1.1). We emphasize that the structured matrix operations in this paper are also useful in many other problems, in addition to condition estimation. Condition estimation for the eigenvalues of a companion matrix C can be used to assess the accuracy of polynomial roots including multiple or clustered roots.

1.2. Overview and notation. The rest of this paper is organized as follows. Section 2 reviews SCE for general eigenvalue problems and gives some new improvements. The fast structured condition estimation scheme is presented in section 3 in detail. Related matrix algorithms are derived. We also briefly discuss the extension of the techniques to general matrices (1.1) in the above class. Section 4 provides the algorithm, together with detailed flop counts, and shows some numerical examples. We draw some concluding remarks in section 6.

The following notation is used in this paper:

- The ith row (or block row) and the jth column (or block column) of $A \equiv (A_{ij})_{n\times n}$ are denoted by $A_{i,:}$ and $A_{:,j}$, respectively. Similarly, $A_{1:i,1:j}$ denotes the submatrix of A at (block) rows 1 through i and (block) columns 1 through j.
- $\mathrm{vec}(A)$ denotes the column vector formed by stacking the columns of A from left to right.
- If A is an SSS matrix, $\mathcal{D}_i(A)$, $\mathcal{U}_i(A)$, etc. represent its SSS generators as in (1.3).
- δA means the product of a small scalar δ with A.
- If a vector u is selected uniformly and randomly from the unit n-sphere S_{n-1}, we write $u \in \mathrm{U}(S_{n-1})$.

2. Condition estimation for general eigenvalue problems.

2.1. General SCE scheme for average (mean) eigenvalues. For a general $n \times n$ real matrix C, Gudmundsson, Kenney, and Laub derived an SCE condition estimator for its average or mean eigenvalues in the following way [19]. Assume we have a block Schur decomposition of C,

$$C = UTU^T, \quad U = [U_1, U_c], \quad T = \begin{bmatrix} T_1 & H \\ 0 & T_c \end{bmatrix}, \tag{2.1}$$

where U is orthogonal and T_1 and T_c have orders n_1 and $n - n_1$, respectively. The average eigenvalue of T_1 is defined to be [1], [19]

$$\mu(T_1) = \frac{\mathrm{trace}(T_1)}{n_1}.$$

If the spectra of T_1 and T_c are well separated [22], [31], then the sensitivity of $\mu(T_1)$ is well defined. A condition number κ for $\mu(T_1)$ is given in [19].

In SCE, a condition estimate for $\mu(T_1)$ can be obtained by perturbing C to $C + \delta E$ with a relative perturbation matrix δE, where δ is a small number, and $E = (C_{ij}Z_{ij})_{n\times n}$ with $Z = (Z_{ij})_{n\times n}$ satisfying $\mathrm{vec}(Z) \in \mathrm{U}(S_{n^2-1})$. Accordingly, $\mu(T_1)$ is perturbed to [19]

$$\mu(\tilde{T_1}) \approx \mu(T_1 + \delta B) = \mu(T_1) + \delta\mu(B), \tag{2.2}$$

where

$$B = U_1^T E U_1 + Y U_c^T E U_1, \tag{2.3}$$

and Y is an $n_1 \times (n - n_1)$ matrix satisfying a Sylvester equation

$$T_1 Y - Y T_c = H. \tag{2.4}$$

Based on (2.2), SCE leads to a relative condition estimate to $\mu(T_1)$ in the following form:

$$\nu = \frac{1}{\omega_p |\mu(T_1)|} |\mu(B)|, \tag{2.5}$$

where p is the number of parameters that define C (for a general $n \times n$ matrix, $p = n^2$; for the companion matrix (1.2), $p = n$), and ω_p is the Wallis factor which can be approximated by [23]

$$\omega_p \approx \sqrt{\frac{2}{\pi(p - \frac{1}{2})}}.$$

The expected value of the estimate $\mathrm{E}(\nu)$ is equal to the exact condition number κ [19].

Multiple samples of Z can be used to increase the accuracy of the estimation. For example, assume we use m samples of Z, denoted $Z^{(i)},\ i = 1, 2, \ldots, m$, which are properly orthonormalized [23], and accordingly, T_1 is perturbed to $T_1 + \delta B^{(i)},\ i = 1, 2, \ldots, m$. Then the m-sample condition estimator is defined as

$$\nu^{(m)} = \frac{1}{\omega_p |\mu(T_1)|} \sqrt{[\mu(B^{(1)})]^2 + \cdots + [\mu(B^{(m)})]^2}.$$

The accuracy of this estimator is given by [19]

$$\Pr\left(\frac{\kappa}{\gamma} \le \nu^{(m)} \le \gamma\kappa\right) \ge 1 - \frac{1}{m!}\left(\frac{2m}{\gamma\pi}\right)^m + \mathrm{O}\left(\frac{1}{\gamma^{m+1}}\right), \qquad \gamma > 1.$$

For example, with $m = 2$, the probability of $\nu^{(m)}$ being within a factor of $\gamma = 10$ of the exact condition number κ is greater than 0.9919. Even with only one sample, this probability is greater than 0.9363.

2.2. Improvements. We make several improvements over Gudmundsson, Kenney, and Laub's general SCE estimator for average eigenvalues. First, for simple real eigenvalues and complex eigenpairs, more specific forms based on (2.3) and (2.5) can be derived. When T_1 is a 1×1 block (eigenvalue), B in (2.3) is reduced to a scalar which can be calculated by using Y (a vector) and the first row and the first column of U. When T_1 is a 2×2 block, T_1 has a conjugate pair of complex eigenvalues. The condition number of this eigenpair is generally different from the condition number of their average. Thus, (2.5) may not precisely reflect the sensitivity of this eigenpair. In fact, by assuming

$$T_1 \equiv \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}, \qquad B \equiv \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix},$$

we can derive a more accurate estimator for the actual condition of the eigenpair [26]

$$\nu = \frac{1}{\omega_p \sqrt{\det(T_1)}} \sqrt{\frac{\alpha^2 \det(T_1) - \alpha\beta \mathrm{trace}(T_1) + \beta^2}{[\mathrm{trace}(T_1)]^2 - 4\det(T_1)}},$$

where $\alpha = b_{11} + b_{22},\ \beta = t_{11}b_{22} + t_{22}b_{11} - t_{12}b_{21} - t_{21}b_{12}$.

The second improvement is that, for average eigenvalues corresponding to a diagonal block T_i other than T_1, we can employ diagonal block swapping techniques as in

[1], [12], [27] to obtain a new Schur decomposition such that T_i (in its similar form) appears in the leading (upper left) position of the new Schur form

$$\tilde{T} = GTG^T, \tag{2.6}$$

where G is an orthogonal transformation matrix.

Another improvement is to rewrite (2.3) as

$$\begin{aligned} B &= \begin{bmatrix} I_{n_1} & Y \end{bmatrix} U^T E U_1 \\ &= \begin{bmatrix} I_{n_1} & Y \end{bmatrix} (U^T E U) \begin{bmatrix} I_{n_1} \\ 0 \end{bmatrix}, \end{aligned} \tag{2.7}$$

where E is isolated from Y. The new representation (2.7) indicates that the operations involving E can be independent of different eigenvalues. In order to compute B for different eigenvalues, we can precompute the matrix $U^T E U$ just once. Then for different eigenvalues we need only solve for Y and then compute the trace of the left $n_1 \times n_1$ submatrix of

$$\hat{B} = \begin{bmatrix} I_{n_1} & Y \end{bmatrix} (U^T E U), \tag{2.8}$$

where appropriate transformations may be applied to $U^T E U$ (see section 4). On the other hand, if multiple samples of E are used, then we can reuse the matrices Y and U. This will be further discussed in section 4.

Finally, for structured matrices C, the perturbation matrix E may also be structured, and it is also possible to compute Y and B quickly by taking advantage of the structure of C. The total cost of the condition estimation can then be less than $O(n^3)$. This is the actual situation for the class of matrices (1.1) where only $O(n^2)$ flops are needed. We elaborate on this in the rest of this paper.

Remark 2.1. The algorithm in this paper can be used to estimate the condition of polynomial roots. For the companion matrix (1.2), it can be shown that the exact condition number κ for $\mu(T_1)$ is actually the condition number defined in [14], [15] for the roots of the polynomial $\sum_{i=0}^{n} a_i x^i$ (with $a_n \equiv 1$) when all roots are distinct [26].

3. Fast structured condition estimation. For a structured matrix C in (1.1), the perturbation matrix E generally corresponds to the low rank modifications (see, e.g., [18] for an error analysis based on perturbing the low rank part of a diagonal plus rank one matrix). In such a situation, the relative perturbation matrix E has the form

$$E = xE_2^T + E_1 y^T = \begin{bmatrix} x & E_1 \end{bmatrix} \begin{bmatrix} E_2^T \\ y^T \end{bmatrix} \equiv \hat{x}\hat{y}^T, \tag{3.1}$$

where $E_1 = (x_{ij} Z_{ij}^x)_{n\times k}$, $E_2 = (y_{ij} Z_{ij}^y)_{n\times k}$ with Z^x and Z^y random matrices satisfying $\mathrm{vec}([Z^x, Z^y]) \in \mathrm{U}(S_{2nk-1})$. For the example of the companion matrix (1.2), E can be further simplified to

$$E = \begin{bmatrix} e^T \\ 0 \end{bmatrix} \begin{matrix} 1 \\ n-1 \end{matrix}, \tag{3.2}$$

where $e^T = [-a_{n-1}z_{n-1}, -a_{n-2}z_{n-2}, \ldots, -(a_0 \mp 1)z_0]$ and $[z_{n-1}, \ldots, z_0]^T \in \mathrm{U}(S_{n-1})$. This is because, usually, $a_{n-1}, a_{n-2}, \ldots, a_0$ are the parameters of interest.

The special structure of E saves the cost of computing B. Moreover, based on the rank structure of C and its similarity transformations, all the major steps in the SCE scheme can be quickly done by structured matrix computations. They include

1. structured Schur decomposition in (2.1),
2. structured diagonal block swapping in (2.6),
3. structured Sylvester equation solution for (2.4), and
4. evaluation of $\mu(B)$ in (2.5) using the low rank structure.

We discuss the details in the following subsections.

3.1. Structured Schur decomposition. We can find a structured Schur decomposition of C by using the fast structured eigensolver in [11]. The traditional Hessenberg QR iterations for the eigenvalues of C are

$$C^{(0)} = C,$$
$$C^{(k)} = Q^{(k)}R^{(k)},\ C^{(k+1)} = R^{(k)}Q^{(k)}, \qquad k = 0, 1, 2, \ldots.$$

Clearly, any $C^{(k)}$ is a rank one update to an orthogonal matrix since C is. The QR algorithm in [11] is based on the result that each $C^{(k)}$ actually has small off-diagonal ranks.

THEOREM 3.1 (see [3], [4], [11]). *The ranks of all off-diagonal blocks* $C^{(k)}_{1:j,j+1:n}$, $j = 1, \ldots, n-1$, *of* $C^{(k)}$ *are no larger than* 3.

The algorithm in [11] uses a sequence of Givens matrices for $Q^{(k)}$ and an SSS form $R^{(k)}$. When the algorithm converges, it yields a quasi-triangular Schur form T. Appropriate Givens matrices form an orthogonal matrix U such that $C = UTU^T$, which is a structured form of (2.1). That is, U is a product of $\mathrm{O}(n^2)$ Givens matrices in general, and T is an SSS matrix.

Clearly, T also has off-diagonal ranks bounded by 3. However, we can compute the QR factorization $T = QR$, where Q is a block diagonal matrix with 1×1 or 2×2 diagonal blocks and R is also a rank one update to an orthogonal matrix. The matrix R has maximum off-diagonal rank bounded by 2 with a proof similar to that of Theorem 3.1 [3], [11]. The application of Q to R does not increase this rank. Thus, we have the following result.

THEOREM 3.2. *For any quasi-triangular matrix orthogonally similar to* C, *its maximum off-diagonal rank is no larger than* 2.

The algorithm in [11] can be used to obtain a compact SSS form of T with maximum generator size 3. This algorithm can be improved by using the techniques in subsection 3.3.2 so that the maximum generator size of T is 2.

3.2. Structured Sylvester equation solver. The Sylvester equation (2.4) can be solved in different ways. It can be converted into a linear system using Kronecker products. Alternatively, since T_1 and T_c are quasi triangular, the Bartels–Stewart algorithm [2] can be applied conveniently. However, both methods cost $\mathrm{O}(O(n_1(n-n_1)^2 + n_1^2(n-n_1)))$ if T_1 and T_c are general quasi-triangular matrices. This makes the total condition estimation cost as large as $\mathrm{O}(n^3)$. Notice that T_1 and T_c are also SSS matrices. We can reduce the Sylvester equation solution cost to $\mathrm{O}(n_1(n-n_1))$ by a structured form of the Bartels–Stewart algorithm. This consists of three parts: making one coefficient matrix lower quasi triangular, quickly formulating certain (simpler) linear or Sylvester systems in the Bartels–Stewart algorithm, and quickly solving those systems.

3.2.1. Transforming the Sylvester equation. We first make one of the two coefficient matrices lower quasi triangular. Since T_1 is block upper triangular, we employ a permutation matrix P such that

$$L \equiv PT_1P^T$$

is lower quasi triangular. The matrix P is simply the anti-identity matrix

$$P = \begin{bmatrix} 0 & & 1 \\ & \iddots & \\ 1 & & 0 \end{bmatrix}.$$

Then (2.4) can be written as

$$L(PY) - (PY)T_c = PH. \tag{3.3}$$

It is clear that the SSS generators of L can be obtained directly from those of T_1 as follows:

$$\mathcal{D}_i(L) = \mathcal{D}_i(T_1)^T, \quad \mathcal{P}_i(L) = \mathcal{V}_i(T_1), \quad \mathcal{Q}_i(L) = \mathcal{U}_i(T_1), \quad \mathcal{R}_i(L) = \mathcal{W}_i(T_1)^T.$$

For notational convenience, we write (3.3) as the SSS Sylvester equation

$$LX - XR = K, \tag{3.4}$$

where L is a block lower triangular compact SSS matrix with generators $\{\mathcal{D}_i(L) \equiv L_{ii}\}_1^{l_1}$, $\{\mathcal{P}_i\}_2^{l_1}$, $\{\mathcal{Q}_i\}_1^{l_1-1}$, and $\{\mathcal{R}_i\}_2^{l_1-1}$, and R is a block upper triangular compact SSS matrix with generators $\{\mathcal{D}_i(R) \equiv R_{ii}\}_1^{l_2}$, $\{\mathcal{U}_i\}_1^{l_2-1}$, $\{\mathcal{V}_i\}_2^{l_2}$, and $\{\mathcal{W}_i\}_2^{l_2-1}$. Here, $l_1 + l_2 = l$ is the total number of diagonal blocks (eigenvalue clusters) in T.

3.2.2. Formulating (simpler) systems in the Bartels–Stewart algorithm. Next, we apply the Bartels–Stewart algorithm to (3.4). Since H is an off-diagonal block of T, we assume $K = PH$ has the form

$$\begin{aligned} K &= \begin{bmatrix} K_{:,1} & K_{:,2} & \cdots & K_{:,l_2} \end{bmatrix} \\ &= \begin{bmatrix} u_1 w_2 \cdots w_{l_1} v_{l_1+1}^T & u_1 w_2 \cdots w_{l_1+1} v_{l_1+2}^T & \cdots & u_1 w_2 \cdots w_{l-1} v_l^T \\ \vdots & \vdots & & \vdots \\ u_{l_1-1} w_{l_1} v_{l_1+1}^T & u_{l_1-1} w_{l_1} w_{l_1+1} v_{l_1+2}^T & \cdots & u_{l_1-1} w_{l_1} \cdots w_{l-1} v_l^T \\ u_{l_1} v_{l_1+1}^T & u_{l_1} w_{l_1+1} v_{l_1+2}^T & \cdots & u_{l_1} w_{l_1+1} \cdots w_{l-1} v_l^T \end{bmatrix}. \end{aligned} \tag{3.5}$$

We also assume that all the matrices in (3.4) have conformal partitions. When we get the solution X of (3.4), the solution of (2.4) can be simply obtained by $Y = P^T X$.

The Bartels–Stewart algorithm solves (3.4) by successively solving

$$\begin{aligned} L_{ii} X_{ij} - X_{ij} R_{jj} = K_{ij} - \sum_{k=1}^{i-1} L_{ik} X_{kj} + \sum_{k=1}^{j-1} X_{ik} R_{kj}, \\ i = 1, 2, \ldots, l_1, \qquad j = 1, 2, \ldots, l_2, \end{aligned}$$

where L_{ii} and R_{jj} are 1×1 or 2×2 blocks, and X_{ij} denotes the (i, j) block of X, which is partitioned conformally according to the blocks of L and R. These equations can be rewritten as a set of linear equations or (simpler) Sylvester equations

$$\begin{aligned} LX_{:,j} - X_{:,j} R_{jj} = \hat{K}_j, \\ j = 1, 2, \ldots, l_2, \end{aligned} \tag{3.6}$$

where $\hat{K}_j = K_{:,j} + X_{:,1:j-1} R_{1:j-1,j}$, and $X_{:,j}$ has one or two columns. Since L is an $n_1 \times n_1$ SSS matrix and R_{jj} is a 1×1 or 2×2 block, we can solve (3.6) for each j in

$O(n_1)$ flops, provided that the right-hand side $\hat{K}_j$ can be evaluated in $O(n_1)$ flops. In fact, the evaluation of both $K_{:,j}$ and $X_{:,1:j-1}R_{1:j-1,j}$ for all $j = 1, \ldots, l_2$ can be done successively as follows (when $j = 1$, let $X_{:,1:j-1}R_{1:j-1,j} \equiv 0$).

For $K_{:,j}$, $j = 1, \ldots, l_2$ in (3.5), introduce auxiliary matrices Ω_k defined by

$$\Omega_{l_1} = I,\ \Omega_k = w_{k+1}\Omega_{k+1}, \qquad k = l_1 - 1,\ l_1 - 2, \ldots, 2, 1. \tag{3.7}$$

Then compute each block K_{kj} by

$$K_{kj} = u_k \Omega_k v_j^T, \qquad k = 1, 2, \ldots, l_1. \tag{3.8}$$

After the calculation of each $K_{:,j}$, replace all Ω_k by

$$\hat{\Omega}_k = \Omega_k w_{l_1+j}. \tag{3.9}$$

For $X_{:,1:j-1}R_{1:j-1,j}$, $j = 1, \ldots, l_2$, notice that the block column $R_{1:j-1,j}$ has the following form:

$$\begin{bmatrix} \mathcal{U}_1\mathcal{W}_2\cdots\mathcal{W}_{j-1}\mathcal{V}_j^T \\ \vdots \\ \mathcal{U}_{j-2}\mathcal{W}_{j-1}\mathcal{V}_j^T \\ \mathcal{U}_{j-1}\mathcal{V}_j^T \end{bmatrix}.$$

Introduce auxiliary matrices Φ_j defined by

$$\Phi_0 = 0,\ \Phi_j = \Phi_{j-1}\mathcal{W}_j + X_{:,j}\mathcal{U}_j, \qquad j = 1, 2, \ldots, l_2 - 1. \tag{3.10}$$

Then clearly,

$$X_{:,1:j-1}R_{1:j-1,j} = \Phi_{j-1}\mathcal{V}_j^T, \qquad j = 1, 2, \ldots, l_2. \tag{3.11}$$

It can be shown that the cost of evaluating $\hat{K}_j$ in (3.6) for each j by (3.7)–(3.11) is $O(n_1)$.

3.2.3. Solving (3.6). Finally, we consider the solution of (3.6). For each j, when R_{jj} is a scalar, (3.6) is an order-n_1 lower triangular SSS system

$$(L - R_{jj})X_{:,j} = \hat{K}_j^T.$$

The coefficient matrix $L - R_{jj}$ has the same generators as L except that the $\mathcal{D}_i$ generators are replaced by $L_{ii} - R_{jj}$ or $L_{ii} - R_{jj}I_2$, depending on the size of L_{ii}. This system can be solved in linear time by the fast SSS system solver in [9], and the details are shown in [26].

When R_{jj} is 2×2, (3.6) is a simple Sylvester equation, which can be converted into an order-$2n$ lower triangular SSS system. Note that for this situation, the Bartels–Stewart algorithm does not apply to (3.6) anymore since we want to maintain real arithmetic and R_{jj} does not have a real Schur form. However, we can rewrite (3.6) as a Sylvester equation in terms of $X_{:,j}^T$ as follows:

$$-R_{jj}^T X_{:,j}^T + X_{:,j}^T L^T = \hat{K}_j^T.$$

This equation can be converted into a lower triangular SSS system

$$(L \otimes I_2 - I_{n_1-2} \otimes R_{jj}^T)\mathrm{vec}(X_{:,j}^T) = \mathrm{vec}(\hat{K}_j^T).$$

The SSS generators of the coefficient matrix are given by those of $L \otimes I_2$, except the diagonal generators are $\mathcal{D}_i(L \otimes I_2) - R_{jj}^T$ or $\mathcal{D}_i(L \otimes I_2) - I_2 \otimes R_{jj}^T$, depending on whether the order of $\mathcal{D}_i(L)$ is 1 or 2. The generators of $L \otimes I_2$ are listed in Table 3.1.

For both cases, the solution of (3.6) costs $O(n_1)$ for each j.

TABLE 3.1

SSS generators of $L\otimes I_2$ in terms of the generators of L, where $\mathcal{P}_i(L) \equiv \begin{bmatrix} \mathcal{P}_{i,1}(L) \\ \mathcal{P}_{i,2}(L) \end{bmatrix}$ and $\mathcal{Q}_i(L) \equiv \begin{bmatrix} \mathcal{Q}_{i,1}(L) \\ \mathcal{Q}_{i,2}(L) \end{bmatrix}$.

Order of $\mathcal{D}_i(L)$	$\mathcal{D}_i(L\otimes I_2)$	$\mathcal{P}_i(L\otimes I_2)$	$\mathcal{Q}_i(L\otimes I_2)$	$\mathcal{R}_i(L\otimes I_2)$
1	$\mathcal{D}_i(L)\otimes I_2$	$I_2\otimes\mathcal{P}_i(L)$	$I_2\otimes\mathcal{Q}_i(L)$	$I_2\otimes\mathcal{R}_i(L)$
2	$\mathcal{D}_i(L)\otimes I_2$	$\begin{bmatrix} I_2\otimes\mathcal{P}_{i,1}(L) \\ I_2\otimes\mathcal{P}_{i,2}(L) \end{bmatrix}$	$\begin{bmatrix} I_2\otimes\mathcal{Q}_{i,1}(L) \\ I_2\otimes\mathcal{Q}_{i,2}(L) \end{bmatrix}$	$I_2\otimes\mathcal{R}_i(L)$

3.3. Swapping the diagonal blocks of the Schur form T. In order to use (2.5) to evaluate the condition of any eigenvalue cluster corresponding to diagonal blocks other than T_1, we can use a swapping procedure to bring those blocks to the leading upper left position of T. Assume that the eigenvalue cluster of interest corresponds to the diagonal blocks $\{T_{i_1}, T_{i_2}, \ldots, T_{i_k}\}$ of T. The matrix T will be transformed into $\tilde{T}$ in (2.6), for which we will derive a new compact SSS form.

3.3.1. Swapping procedure for contiguous blocks. We make use of a fundamental swapping procedure in [1], [12], [27] for two diagonal blocks of a matrix

$$\begin{bmatrix} T_i & H_i \\ 0 & T_j \end{bmatrix},$$

where T_i and T_j have orders n_i and n_j, respectively, and T_i and T_j have no eigenvalue in common. The swapping procedure in [1], [12], [27] finds an orthogonal matrix G_i, which is the product of some Givens matrices such that

$$G_i \begin{bmatrix} T_i & H_i \\ 0 & T_j \end{bmatrix} G_i^T = \begin{bmatrix} M_j T_j M_j^{-1} & \bar{H}_i \\ 0 & M_i T_i M_i^{-1} \end{bmatrix},$$

where M_i and M_j are approximate invertible matrices. Thus T_i and T_j have been swapped.

In order to preserve the quasi-triangular form of T, we apply this swapping procedure to contiguous 1×1 or 2×2 diagonal blocks of T, even if T may have multiple eigenvalues. The details are as follows. To bring the 1×1 or 2×2 blocks $\{T_{i_1}, T_{i_2}, \ldots, T_{i_k}\}$ of an eigencluster to the leading position, we partition T as

$$T = \begin{bmatrix} \hat{T}_1 & \hat{H}_1 & \cdots \\ 0 & \hat{T}_2 & \cdots \\ 0 & 0 & \ddots \end{bmatrix},$$

where $\hat{T}_2$ has diagonal blocks $\{T_{i_1}, T_{i_2}, \ldots, T_{i_k}\}$. If we directly apply the above swapping procedure to $\begin{bmatrix} \hat{T}_1 & \hat{H}_1 \\ 0 & \hat{T}_2 \end{bmatrix}$ to bring $\hat{T}_2$ to the leading position, then the quasi-triangular form of T may be destroyed, and also the structures of the related matrices and matrix equations are hard to explore.

Thus, instead, we apply the above procedure to contiguous 1×1 or 2×2 diagonal blocks of T and bring each T_i in $\{T_{i_1}, T_{i_2}, \ldots, T_{i_k}\}$ to the leading position in one round of swapping. After each round of swapping, T is transformed into a new quasi-triangular matrix $\tilde{T} = GTG^T$ as in (2.6), where G is a product of Givens matrices. (The current structure of this $\tilde{T}$ needs to be compressed. See the next subsection.) The number of Givens matrices depends on the size of T_i. If T_i is 1×1, then $k_i - 1$

Givens matrices are needed, where k_i is the row or column index of T_i in T. The matrix G has the form

$$G = \prod_{j=1}^{k_i-1} \operatorname{diag}\left[I_{j-1}, \begin{bmatrix} c_j & s_j \\ -s_j & c_j \end{bmatrix}, I_{n-j-1}\right], \tag{3.12}$$

which is upper Hessenberg. If T_i is 2×2, then $2(k_i - 1)$ Givens matrices are needed, where k_i is the row or column index of the leading entry of T_i in T. Some of these Givens matrices commute, and after reordering the matrices, we have $G = G_1 G_2$, where each of G_1 and G_2 has the form (3.12). The details of the generation of G are similar to those in [26]. Clearly, G in (3.12) is an SSS matrix, with its generators given in Table 3.2 in terms of its Schur parameters $\{c_i, s_i\}$ [11].

TABLE 3.2
SSS generators of G in (3.12).

$\mathcal{D}_j(G)$	$\mathcal{U}_j(G)$	$\mathcal{V}_j(G)$	$\mathcal{W}_j(G)$	$\mathcal{P}_j(G)$	$\mathcal{Q}_j(G)$	$\mathcal{R}_j(G)$
$c_{j-1}c_j$	$c_{j-1}s_j$	c_j	s_j	1	$-s_j$	0

The matrix $\tilde{T}$ is then still quasi triangular. We can get its SSS form by multiplying three SSS matrices in (2.6) using the formulas for SSS matrix multiplications in [8]. This is the method used in [26] for the situation of distinct eigenvalues. Notice that the off-diagonal generator sizes increase accumulatively with the multiplications in (2.6). If the $\mathcal{W}_i(T)$ generators have size 2, then the $\mathcal{W}_i(\tilde{T})$ generators have size up to 6, since the $\mathcal{W}_i$ generators of G and G^T have size 1 or 2.

3.3.2. Recovery of compact SSS representation of $\tilde{T}$. Since here we are considering general eigenclusters instead of simple distinct eigenvalues, the matrix multiplication technique in [26] is inefficient. For example, if the swapping process is applied to an eigencluster which has k multiple eigenvalues or eigenpairs, then $\tilde{T}$ needs to be multiplied by up to $\mathrm{O}(kn)$ Givens matrices, and the off-diagonal generator sizes of $\tilde{T}$ increase significantly. Therefore, $\tilde{T}$ is generally not compact anymore. On the other hand, the actual off-diagonal ranks of $\tilde{T}$ do not increase, according to Theorem 3.2. Thus, we use a recovery strategy similar to the one in [11] to reconstruct a compact SSS form for $\tilde{T}$. The recovery strategy in [11] is designed for certain strictly triangular matrices with maximum off-diagonal rank 2 such that their generator sizes are bounded by 3. Here we improve the strategy such that it works for the quasi-triangular matrix $\tilde{T}$, and furthermore, the generators sizes are bounded by 2, which is consistent with the maximum off-diagonal rank. This also saves one SSS matrix multiplication.

The matrix T is orthogonally similar to C and is obviously orthogonal plus rank one, which we assume to be $T \equiv P + uv^T$. Also, assume that $\tilde{T}$ has a QR factorization $\tilde{T} = \tilde{Q}\tilde{R}$. The matrix $\tilde{Q}$ is a block diagonal matrix with 1×1 or 2×2 diagonal blocks, since $\tilde{T}$ is quasi triangular. We have

$$\begin{aligned} \tilde{R} &= \tilde{Q}^T \tilde{T} = \tilde{Q}^T G(P + uv^T)G^T \\ &= \tilde{Q}^T GPG^T + (\tilde{Q}^T Gu)(Gv)^T \equiv \tilde{P} + \tilde{u}\tilde{v}^T. \end{aligned}$$

There exists an orthogonal upper Hessenberg matrix P_1 which is a product of Givens matrices such that

$$P_1 \tilde{u} = ||\tilde{u}||_2 e_1, \tag{3.13}$$

TABLE 3.3

Upper SSS generators of $P_1^T P_2$ in terms of the Schur parameters of P_1 and P_2, where $\rho_{i-1} \equiv \prod_{k=1}^{i-1} s_k \tilde{s}_k + \sum_{j=1}^{i-1} (c_j \tilde{c}_j \prod_{k=j+1}^{i-1} s_k \tilde{s}_k)$. Other generators can be obtained by symmetry in the structure.

$D_i(P_1^T P_2)$	$U_i(P_1^T P_2)$	$V_i(P_1^T P_2)$	$W_i(P_1^T P_2)$
$c_i \tilde{c}_i \rho_{i-1} + s_i \tilde{s}_i$	$c_i \tilde{s}_i \rho_{i-1} - s_i \tilde{c}_i$	$\tilde{c}_i$	$\tilde{s}_i$

TABLE 3.4

SSS generators of $\tilde{u}\tilde{v}^T$.

$\mathcal{D}_i(\tilde{u}\tilde{v}^T)$	$\mathcal{U}_i(\tilde{u}\tilde{v}^T)$	$\mathcal{V}_i(\tilde{u}\tilde{v}^T)$	$\mathcal{W}_i(\tilde{u}\tilde{v}^T)$	$\mathcal{P}_i(\tilde{u}\tilde{v}^T)$	$\mathcal{Q}_i(\tilde{u}\tilde{v}^T)$	$\mathcal{R}_i(\tilde{u}\tilde{v}^T)$
$\tilde{u}_i \tilde{v}_i^T$	$\tilde{u}_i$	$\tilde{v}_i$	1	$\tilde{u}_i$	$\tilde{v}_i$	1

where e_1 is the first unit vector. Then we have that $P_1 \tilde{P} = P_1 \tilde{R} - ||\tilde{u}||_2 e_1 \tilde{v}^T$ is orthogonal and also upper Hessenberg. Thus, there exists another orthogonal upper Hessenberg matrix P_2 such that

$$P_2 = P_1 \tilde{P} = P_1 \tilde{R} - ||\tilde{u}||_2 e_1 \tilde{v}^T, \tag{3.14}$$

$$\tilde{R} = P_1^T P_2 + ||\tilde{u}||_2 P_1^T e_1 \tilde{v}^T = P_1^T P_2 + \tilde{u}\tilde{v}^T, \tag{3.15}$$

$$\tilde{T} = \tilde{Q}\tilde{R} = \tilde{Q}(P_1^T P_2 + \tilde{u}\tilde{v}^T). \tag{3.16}$$

Both P_1 and P_2 are orthogonal upper Hessenberg and have maximum off-diagonal rank one, and $||\tilde{u}||_2 P_1^T e_1 \tilde{v}^T$ is a rank one matrix. Therefore, if SSS multiplication and addition formulas are used as in [11], the SSS form of $\tilde{R}$ has maximum generator size 3. However, a more compact form of $\tilde{R}$ is available.

THEOREM 3.3. *For any orthogonal upper Hessenberg matrices P_1 and P_2, the matrix $P_1^T P_2$ has maximum off-diagonal rank one.*

Proof. Denote a submatrix $P_1(1:i, 1:i+1)$ by $P_{1;(1:i,1:i+1)}$. An upper off-diagonal block of $P = P_1^T P_2$ is given by

$$P_{1:i,i+1:n} = P_{1;(1:i+1,1:i)}^T P_{2;(1:i+1,i+1:n)}.$$

The submatrix $P_{2;(1:i+1,i+1:n)}$ has rank one, since all its rows are multiples of

$$[\tilde{c}_i,\ \tilde{s}_i \tilde{c}_{i+1},\ \ldots,\ \tilde{s}_i \cdots \tilde{s}_{n-1} \tilde{c}_n],$$

where $\{\tilde{c}_i, \tilde{s}_i\}$ are the Schur parameters of P_2. (See Table 3.2.) Thus, $P_{1:i,i+1:n}$ has rank one also. ∎

The SSS form of $P_1^T P_2$ in (3.16) is given in Table 3.3 in terms of the Schur parameters $\{c_i, s_i\}$ and $\{\tilde{c}_i, \tilde{s}_i\}$ of P_1 and P_2, respectively. The generators can be obtained in O(n) complexity by computing $\rho_{i-1} = \prod_{k=1}^{i-1} s_k \tilde{s}_k + \sum_{j=1}^{i-1} \left(c_j \tilde{c}_j \prod_{k=j+1}^{i-1} s_k \tilde{s}_k \right) \equiv \alpha_{i-1} + \beta_{i-1}$ recursively as follows:

$$\begin{aligned} &\alpha_0 = 1,\ \alpha_i = (s_i \tilde{s}_i)\alpha_{i-1}, \qquad i = 1, 2, \ldots, n-1, \\ &\beta_0 = 0,\ \beta_i = (s_i \tilde{s}_i)\beta_{i-1} + c_i \tilde{c}_i, \qquad i = 1, 2, \ldots, n-1. \end{aligned}$$

The SSS form of $\tilde{u}\tilde{v}^T$ is given in Table 3.4. Then one SSS addition gives an SSS form for $\tilde{R}$ whose maximum generator size is 2.

The left multiplication of $\tilde{R}$ by $\tilde{Q}$ does not increase the off-diagonal block ranks because $\tilde{Q}$ is a block diagonal matrix with 1×1 or 2×2 diagonal blocks. Thus, $\tilde{T}$ has

maximum generator size 2. This construction process provides an alternative way of proving Theorem 3.2.

Therefore, we can use (3.14)–(3.16) to recover a compact SSS form for $\tilde{T}$. After each round of swapping to bring a single eigenvalue or eigenpair to a desired position, we apply the recovery procedure to $\tilde{T}$. First, we form the redundant SSS form of $\tilde{T}$ in (2.6) by SSS matrix multiplications, and then we compute the QR factorization $\tilde{Q}\tilde{R}$ for $\tilde{T}$. The matrix $\tilde{Q}$ can be obtained by computing the Givens QR factorization

$$\tilde{T}_i = \tilde{Q}_i \tilde{R}_i \tag{3.17}$$

for any 2×2 diagonal block $\tilde{T}_i$ of T. The block diagonal matrix $\tilde{Q}$ has each diagonal block being either 1 or $\tilde{Q}_i$. The matrix $\tilde{R}$ is still an SSS matrix with the same generators as $\tilde{T}$ except that for any i corresponding to a 2×2 diagonal block,

$$\mathcal{D}_i(\tilde{R}) = \tilde{R}_i,\ \mathcal{U}_i(\tilde{R}) = \tilde{Q}_i^T \mathcal{U}_i(\tilde{T}). \tag{3.18}$$

Note that the Schur parameters of P_2 are obtained by updating only certain SSS generators of $P_1\tilde{R}$. With $P_1^T P_2$ and $\tilde{u}\tilde{v}^T$ available in SSS forms, a new compact SSS form of $\tilde{T}$ is straightforward according to (3.17) and (3.18). This improved recovery procedure requires about $14i_1$ operations, as compared with the $40i_1$ cost in [11]. In addition, the SSS generators have maximum size 2 instead of 3, which reduces the cost for later operations also.

3.4. Computing the condition estimate (2.5). The major work in evaluating ν in (2.5) is to compute $\mu(B)$, where B is given by (2.3). In general, for companion matrices we can compute (2.5) using the method which will be presented in section 4. But since E has a special form (3.2) with only one nonzero row, an alternative method is to use (2.3) directly.

We first find $U_{1,:}$. Note that the fast eigensolver in [11] provides U in the form of a sequence of $O(n^2)$ Givens rotation matrices. Thus, the application of these matrices on the right to e_1^T, the first unit vector of length n, yields the initial $U_{1,:}$. This costs $O(n^2)$ operations. Later, for each cluster of diagonal blocks with size n_i, the row $U_{1,:}$ needs to be updated when T is updated by the swapping process. According to the previous subsection, U is updated to $\tilde{U} = UG^T$, where G is also represented by Givens rotation matrices. Thus, the updated vector is

$$\tilde{U}_{1,:} = U_{1,:}G^T. \tag{3.19}$$

The computation of EU_1 in (2.3) can be done by considering EU. (If the diagonal blocks of T are swapped, then we compute $E\tilde{U}$ similarly.) Since U is represented by $O(n^2)$ Givens matrices, the cost for computing EU is $O(n^2)$. The matrix EU_1 has only one nonzero row, which we assume to be u_1^T. Also, let $U_{1,:} = \begin{bmatrix} U_{1:n_1,:} \\ U_{n_1+1:n,:} \end{bmatrix}$. Then we have

$$B = U_{1,1:n_1}^T u_1^T + (YU_{1,n_1+1:n}^T)u_1^T,$$

which is the sum of two rank one matrices. We first evaluate $YU_{1,n_1+1:n}^T$ and then compute the diagonal entries of B.

4. The case of general C in (1.1). For a general C in (1.1), which is a low rank modification to a symmetric, skew-symmetric, or orthogonal matrix, the main operations in previous sections are similar. For example, we can quickly get an SSS

form Schur decomposition. A major difference is that the computation of the condition estimate (2.5) can be done in a more general way.

For C in (1.1), the perturbation matrix E has the form (3.1). We can precompute

$$U^T E U = (U^T \hat{x})(U^T \hat{y})^T \equiv \tilde{x}\tilde{y}^T.$$

The computations of $\tilde{x}$ and $\tilde{y}$ cost $O(n^2)$ flops, since U is a product of $O(n^2)$ Givens rotation matrices and both $\tilde{x}$ and $\tilde{y}$ have a finite number of columns.

The direct computation of the trace of $\hat{B}$ in (2.8) is thus straightforward. Since $\hat{B} = \left(\begin{bmatrix} I_{n_1} & Y \end{bmatrix}\tilde{x}\right)\tilde{y}^T$, we first form $\begin{bmatrix} I_{n_1} & Y \end{bmatrix}\tilde{x}$ and then compute the trace of the left $n_1 \times n_1$ submatrix of $\hat{B}$. For different eigenvalues, permutations are applied to U, and $\hat{B}$ now has the form

$$\hat{B} = \begin{bmatrix} I_{n_1} & Y \end{bmatrix} (G\tilde{x})(G\tilde{y})^T, \tag{4.1}$$

where G is a product of Givens rotation matrices. (Here, Y should also be different, but the same notation is used for convenience.) We form $G\tilde{x}$ and $G\tilde{y}$ first, and the rest of the computations are similar.

5. Algorithm, flop counts, and numerical experiments. We outline the major steps in the following algorithm in terms of a companion matrix C.

ALGORITHM 1 (condition estimation for the eigenvalues/eigenclusters of C).

1. Compute an initial structured Schur decomposition $C = UTU^T$.
2. Choose a perturbation matrix E as in (3.2) or (3.1). Precompute $U^T E U$ as in section 4.
3. Repeat for each eigenvalue cluster i corresponding to $\{T_{i_1}, T_{i_2}, \ldots\}$.
   (a) If $i > 1$, use the swapping technique in subsection 3.3 to bring cluster i to the leading position, one block T_{i_j} per round.
   (b) Solve the Sylvester equation (2.4) as in subsection 3.2.
   (c) Compute the condition estimate (2.5) via the diagonal entries of $\hat{B}$ in (4.1).
4. If additional samples of E are used, repeat steps 2 and 3(c).

5.1. Flop counts. To obtain detailed flop counts for a companion matrix C, we make the following assumptions:

- The number of iterations required for the Hessenberg QR iteration to converge is cn^2, where c is a constant (c is usually small).
- Each compact SSS matrix A has maximum off-diagonal rank p which is 2 for T and 1 for an orthogonal upper Hessenberg matrix. All $\mathcal{W}_i$ and $\mathcal{R}_i$ generators of A have dimension p.
- A simplified problem is considered, where all diagonal blocks T_i of T are 1×1.
- The matrix T has m eigenvalue clusters and the ith cluster has n_i eigenvalues $\{T_{i_1}, T_{i_2}, \ldots\}$.

Step 1 costs about the same as the structured eigensolver in [11], and we do not discuss the details here. Step 2 costs about $6cn^2$ flops. The cost of computing the condition estimate of each cluster i in step 3 is as follows.

1. In step 3(a), the operations and the required flops are given by the following:
   (a) Swapping the diagonals of T to bring T_{i_j} to the leading position and computing a redundant SSS form for $\tilde{T}$ cost
   $$(80p^3 + 122p^2 + 130p + 81)(i_j - j),$$
   where we have used the result that it costs at most $40p^3(i_j - j)$ flops to multiply two order-$(i_j - j)$ SSS matrices whose maximum off-diagonal ranks are p [8].
   (b) Recovery of a compact SSS form for $\tilde{T}$ approximately costs
   $$(47p^3 + 242p^2 + 464p + 374)(i_j - j),$$
   where, for simplicity, the cost for a general rank p recovery process is mainly counted based on SSS multiplications, although it is possible to extend the idea in subsection 3.3.2 to further reduce the cost.

   The total cost for the entire cluster i is thus
   $$(127p^3 + 364p^2 + 594p + 455)\sum_{j=1}^{n_i}(i_j - j).$$
2. The cost for step 3(b) using the Bartels–Stewart algorithm is
   $$(2p^3 + 8p^2 + 11p + 2)n_i(n - n_i).$$
3. Step 3(c) costs
   $$2n_i(n - n_i) + 2n_i + 12\sum_{j=1}^{n_i}(i_j - j).$$

Therefore, the cost for all the eigenvalue clusters is

$$(127p^3 + 364p^2 + 594p + 467)\sum_{i=2}^{m}\sum_{j=1}^{n_i}(i_j - j)$$
$$+ (2p^3 + 8p^2 + 11p + 4)\sum_{i=1}^{m} n_i(n - n_i) + 2\sum_{i=1}^{m} n_i.$$

Since $\sum_{i=1}^{m} n_i = n$, we have

$$\sum_{i=2}^{m}\sum_{j=1}^{n_i}(i_j - j) = \sum_{i=2}^{m} n_i(i_1 - 1) \le n\sum_{i=2}^{m} n_i \le n^2,$$
$$\sum_{i=1}^{m} n_i(n - n_i) \le n\sum_{i=1}^{m} n_i = n^2.$$

The total cost is thus approximately bounded by

$$(129p^3 + 372p^2 + 605p + 471)n^2. \tag{5.1}$$

This bound can highly overestimate the cost. For example, when there are only two eigenvalue clusters with equal size $n/2$, the cost is bounded by

$$\left(33p^3 + 95p^2 + 154p + 119\right)n^2.$$

If multiple samples are used, we need only repeat steps 2 and 3(c), and the results from other steps can be reused. Since the total cost for steps 2 and 3(c) is bounded by $26n^2$, which is much smaller than (5.1), the amount of work required for each additional sample of SCE is insignificant.

5.2. Numerical examples. A MATLAB implementation of Algorithm 1 is available at http://www.math.purdue.edu/~xiaj/work/sceeig.

We apply it to some companion matrices and demonstrate the efficiency and accuracy. Note that [26] also includes some results with a different algorithm which requires all the eigenvalues to be distinct.

Example 5.1. Consider a companion matrix C whose eigenvalues are $\lambda_i = i,\ i = 1, 2, \ldots, n$. These eigenvalues are the roots of the Wilkinson polynomial. According to [26], the SCE estimator (2.5) is an estimate of the following exact condition number for λ_i:

$$\kappa_i = ||(k_{i,1}, k_{i,2}, \ldots, k_{i,n})^T||_2,\ \kappa_{i,j} = \left| \frac{a_j \lambda_i^{j-1}}{\prod_{k \neq i} (\lambda_i - \lambda_k)} \right|, \tag{5.2}$$

where a_j is the coefficient of the λ^j term of the polynomial (see (1.2)).

For $n = 15$, we calculate the exact condition numbers κ_i, their 1-sample SCE estimates, and the estimates by the MATLAB routine `condeig`, which computes the reciprocals of the cosines of the angles between the left and right eigenvectors of C. According to Figure 5.1, SCE provides favorable estimates, while `condeig` gives large estimates for nearly all eigenvalues except the first one.

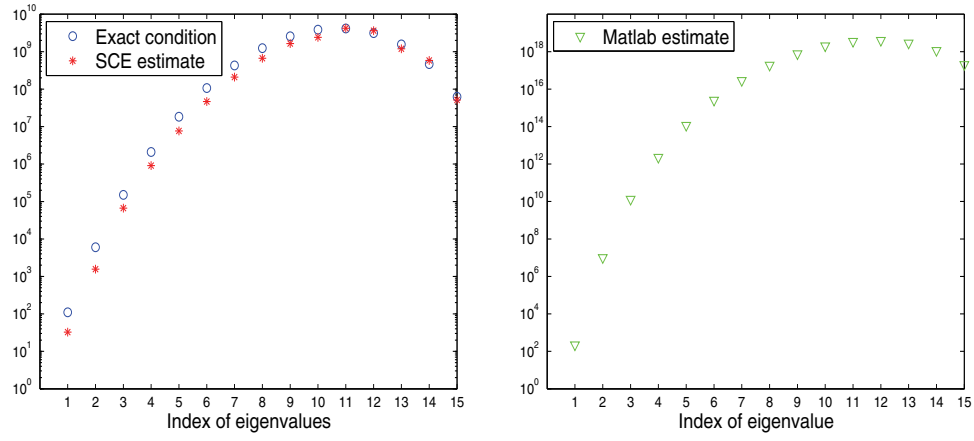


FIG. 5.1. *Condition numbers and their estimates for Example* 5.1.

Example 5.2. We show the quadratic complexity of the estimator with an example where C in (1.2) has $a_0 = 1$ and $a_i = 0,\ i = 1, \ldots, n-1$. We report only the flop counts of our preliminary MATLAB implementation of the algorithm.

This companion matrix has eigenvalues at the roots of unity. The eigenvalues are all well conditioned, with κ_i in (5.2) given by $1/n$ [15]. Our SCE estimator also reflects this fact. We run our algorithm for n ranging from 32 to 1024 and count the flops, denoted flops_n. Then we compute the flop ratios $\frac{\text{flops}_n}{\text{flops}_{n/2}}$. The numerical results

TABLE 5.1
SCE for Example 5.2 *with different* n.

n	32	64	128	256	512	1024
κ_{exact}	0.0313	0.0156	0.0078	0.0039	0.0020	0.0010
κ_{SCE}	0.0432	0.0268	0.0124	0.0083	0.0009	0.0007
$\frac{\text{flops}_n}{\text{flops}_{n/2}}$	4.9	4.4	4.2	4.1	4.0	4.0

TABLE 5.2
SCE for Example 5.4 *with multiplicity* $m = 2$.

n	32	64	128	256	512	1024
$\frac{\text{flops}_n}{\text{flops}_{n/2}}$	4.8	4.4	4.2	4.1	4.0	4.0

show that the ratios are close to 4, which is consistent with the $O(n^2)$ complexity. See Table 5.1.

Example 5.3. We consider a companion matrix C with multiple eigenvalues $\{2^{-i}, 2^{-i}\}$, $i = 1, 2, \ldots, n/2$.

For $n = 10$, SCE gives five estimates for the five eigenvalue clusters $\{2^{-i}, 2^{-i}\}$: $2.5E2$, $9.9E2$, $6.4E1$, $3.4E2$, $4.7E1$. We see that the eigenvalue clusters are still well conditioned. This somehow is consistent with the result that multiple roots of polynomials may be well conditioned if the multiplicities are preserved on a proper pejorative manifold [22], [32].

However, for large problems with multiple eigenvalues, eigensolvers such as the one in [11] give inaccurate results. Thus, it is not clear if SCE with those eigensolvers accurately reflects the condition. With multiplicity preserving eigensolvers, it is possible to further explore the potential of SCE. This remains an open problem. Therefore, in the following example, we report only the complexity results.

Example 5.4. Consider a companion matrix C whose eigenvalues are the roots of unity with different multiplicities m. (We chose this example because its nonzero entries are relatively easy to compute accurately.) Tables 5.2 shows the flop ratios $\frac{\text{flops}_n}{\text{flops}_{n/2}}$ for $m = 2$. Very similar results can be observed for m to be a fraction of n such as $\frac{n}{4}$. Thus we omit them.

6. Conclusions. We develop a condition estimation scheme for the eigenvalues of a class of matrices which are low rank perturbations to rank symmetric matrices. Rank structures of these matrices are exploited and fast structured matrix operations are presented, such as Schur decomposition, matrix equation solution, Schur form update, compact semiseparable form reconstruction, etc. These operations may be used in the condition estimation of other structured matrices and more general problems such as invariant subspace computations.

Similar techniques can also be used to estimate the condition of the eigenvectors. The information in the condition estimation for the eigenvalues can be reused. It is also possible to derive a condition estimate for the average eigenvalue of the block T_c in (2.1). In this way, we can save about 3/4 of the diagonal swapping work, on average, for all the eigenvalues. We also notice that the cost for the structured Sylvester solver can be possibly reduced further, since K in (3.4) is a low rank matrix.

REFERENCES

[1] Z. Bai, J. W. Demmel, and A. McKenney, *On computing condition numbers for the nonsymmetric eigenproblem*, ACM Trans. Math. Software, 19 (1993), pp. 202–223.

[2] R. H. Bartels and G. W. Stewart, *Solution of the matrix equation $AX + XB = C$*, Comm. ACM, 15 (1972), pp. 820–826.

[3] D. Bindel, S. Chandresekaran, J. Demmel, D. Garmire, and M. Gu, *A Fast and Stable Nonsymmetric Eigensolver for Certain Structured Matrices*, Technical report, University of California, Berkeley, CA, 2005.

[4] D. A. Bini, F. Daddi, and L. Gemignani, *On the shifted QR iteration applied to companion matrices*, Electron. Trans. Numer. Anal., 18 (2004), pp. 137–152.

[5] D. A. Bini, Y. Eidelman, L. Gemignani, and I. Gohberg, *Fast QR Eigenvalue Algorithms for Hessenberg Matrices which Are Rank-One Perturbations of Unitary Matrices*, Technical Report 1587, Department of Mathematics, University of Pisa, Pisa, Italy, 2005.

[6] S. P. Chan, R. Feldman, and B. N. Parlett, *A program for computing the condition numbers of matrix eigenvalues without computing eigenvectors*, ACM Trans. Math. Software, 3 (1977), pp. 186–203.

[7] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A.-J. van der Veen, and D. White, *Some fast algorithms for sequentially semiseparable representations*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 341–364.

[8] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A.-J. van der Veen, and D. White, *Fast Stable Solvers for Sequentially Semi-separable Linear Systems of Equations and Least Squares Problems*, Technical report, University of California, Berkeley, CA, 2003.

[9] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, and A.-J. van der Veen, *Fast stable solver for sequentially semi-separable linear systems of equations*, in High Performance Computing (HiPC 2002): 9th International Conference, Lecture Notes in Comput. Sci. 2552, Springer-Verlag, Heidelberg, 2002, pp. 545–554.

[10] S. Chandrasekaran, M. Gu, X. Sun, J. Xia, and J. Zhu, *A superfast algorithm for Toeplitz systems of linear equations*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1247–1266.

[11] S. Chandrasekaran, M. Gu, J. Xia, and J. Zhu, *A fast QR algorithm for companion matrices*, in Recent Advances in Matrix and Operator Theory, Oper. Theory Adv. Appl. 179, Birkhäuser, Basel, 2008, pp. 111–143.

[12] J. J. Dongarra, S. Hammarling, and J. H. Wilkinson, *Numerical considerations in computing invariant subspaces*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 145–161.

[13] H. Fassbender and D. Kressner, *Structured eigenvalue problems*, GAMM-Mitt., 29 (2006), pp. 297–318.

[14] W. Gautschi, *On the condition of algebraic equations*, Numer. Math., 21 (1973), pp. 405–424.

[15] W. Gautschi, *Questions of numerical condition related to polynomials*, in Studies in Numerical Analysis, MAA Studies in Math. 24, G. H. Golub, ed., Math. Assoc. Amer., Washington, DC, 1984, pp. 140–177.

[16] L. Gemignani, *A unitary Hessenberg QR-based algorithm via semiseparable matrices*, J. Comput. Appl. Math., 184 (2005), pp. 505–517.

[17] G. Golub and C. V. Loan, *Matrix Computation*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[18] M. Gu and S. C. Eisenstat, *A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1266–1276.

[19] T. Gudmundsson, C. Kenney, and A. J. Laub, *Small-sample statistical estimates for the sensitivity of eigenvalue problems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 868–886.

[20] T. Gudmundsson, C. S. Kenney, A. J. Laub, and M. S. Reese, *Applications of small-sample statistical condition estimation in control*, in Proceedings of the 1996 IEEE International Symposium on Computer-Aided Control System Design, IEEE Press, Piscataway, NJ, 1996, pp. 164–169.

[21] B. Kågstrom and P. Poromaa, *Computing eigenspaces with specified eigenvalues of a regular matrix pair (A, B) and condition estimation: Theory, algorithms and software*, Numer. Algorithms, 12 (1996), pp. 369–407.

[22] W. Kahan, *Conserving Confluence Curbs Ill-Condition*, Technical Report 6, Department of Computer Science, University of California, Berkeley, CA, 1972.

[23] C. S. Kenney and A. J. Laub, *Small-sample statistical condition estimates for general matrix functions*, SIAM J. Sci. Comput., 15 (1994), pp. 36–61.

[24] C. S. Kenney, A. J. Laub, and M. S. Reese, *Statistical condition estimation for linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 566–583.

[25] C. S. Kenney, A. J. Laub, and M. S. Reese, *Statistical condition estimation for linear least squares*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 906–923.
[26] A. J. Laub and J. Xia, *Statistical condition estimation for the roots of polynomials*, SIAM J. Sci. Comput., 31 (2008), pp. 624–643.
[27] G. W. Stewart, *Algorithm* 506*: HQR*3 *and EXCHNG: Fortran subroutines for calculating and ordering eigenvalues of a real upper Hessenberg matrix*, ACM Trans. Math. Software, 2 (1976), pp. 275–280.
[28] C. Van Loan, *On estimating the condition of eigenvalues and eigenvectors*, Linear Algebra Appl., 88/89 (1987), pp. 715–732.
[29] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1963.
[30] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon, Oxford, UK, 1965.
[31] J. H. Wilkinson, *Sensitivity of eigenvalues*, Utilitas Math., 25 (1984), pp. 5–76.
[32] Z. Zeng, *A method computing multiple roots of inexact polynomials*, in Proceedings of ISSAC 2003, ACM Press, New York, 2003, pp. 266–272.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1677–1696

ORDINAL RANKING FOR GOOGLE'S PAGERANK*

REBECCA S. WILLS† AND ILSE C. F. IPSEN‡

Abstract. We present computationally efficient criteria that can guarantee correct ordinal ranking of Google's PageRank scores when they are computed with the power method (ordinal ranking of a list consists of assigning an ordinal number to each item in the list). We discuss the tightness of the ranking criteria, and illustrate their effectiveness for top k and bucket ranking. We present a careful implementation of the power method, combined with a roundoff error analysis that is valid for matrix dimensions $n < 10^{14}$. To first order, the roundoff error depends neither on n nor on the iteration count, but only on the maximal number of inlinks and the dangling nodes. The applicability of our ranking criterion is limited by the roundoff error from a single matrix vector multiply. Numerical experiments suggest that our criteria can effectively rank the top PageRank scores. We also discuss how to implement ranking for extremely large practical problems, by curbing roundoff error, reducing the matrix dimension, and using faster converging methods.

Key words. ranking distance, power method, stochastic matrix, PageRank, Google matrix, ordinal rank, roundoff error

AMS subject classifications. 62F07, 65F15, 65F50, 65C40, 15A18, 15A51, 68P20

DOI. 10.1137/070698129

1. Introduction.

Google founders Larry Page and Sergey Brin developed the PageRank algorithm primarily for ranking Web pages. In addition to its continued use in many Google Web search tools [1], the PageRank algorithm reaches beyond the Web to many other applications involving directed graphs such as social networks and semantic networks [2, 9, 11, 14, 19, 30, 33, 37, 45], as well as genomics [34], and identifying sources of hospital infections [41]. In fact, Hilgers and Langville recently identified the PageRank algorithm as one of the five greatest applications of Markov Chains [44].

The Google PageRank vector π is the stationary distribution of a $n \times n$ stochastic matrix G

$$\pi^T G = \pi^T, \qquad \pi \geq 0, \qquad \|\pi\|_1 = 1. \tag{G}$$

Each component of π measures the importance of a web page [8, 36]. If $\pi_i > \pi_j$, then web page i has higher PageRank than web page j, and page i may be displayed ahead of page j among the search results.

The matrix G is a convex combination of two stochastic matrices, $G \equiv \alpha S + (1 - \alpha)\mathbf{1}v^T$. Here $0 < \alpha < 1$ is a scalar, which was originally set to .85 [8, section 2.1.1]; S is an $n \times n$ stochastic matrix; $\mathbf{1}$ is the column vector of all ones; and $v \geq 0$ is a column vector with $\|v\|_1 = 1$. Because $\alpha < 1$ and $\mathbf{1}$ is a right eigenvector of S, the eigenvalue one of G is algebraically simple [13, 20, 40], which implies that π is unique.

Row i of S contains the outgoing links from web page i to other pages, while column i contains the incoming links from other pages to page i. A web page without

*Received by the editors July 24, 2007; accepted for publication (in revised form) by N. J. Higham September 18, 2008; published electronically January 23, 2009.
http://www.siam.org/journals/simax/30-4/69812.html
†Department of Mathematics, Computer Science and Physics, Roanoke College, 221 College Lane, Salem, VA 24153 (wills@roanoke.edu, http://faculty.roanoke.edu/wills/).
‡Department of Mathematics, North Carolina State University, P.O. Box 8205, Raleigh, NC 27695-8205 (ipsen@ncsu.edu, http://www4.ncsu.edu/~ipsen/).

any outgoing links (such as a pdf, image, or audio file) is called a *dangling node*. Zero rows corresponding to dangling nodes are replaced by a *dangling node vector* that has nonnegative elements summing to one.

In their 1998 paper [36, section 2.6] Google founders Brin and Page computed PageRank with the power method.

Power method applied to G. Let $x^{(0)} \geq 0$ with $\|x^{(0)}\|_1 = 1$. Repeat

$$[x^{(k+1)}]^T = [x^{(k)}]^T G, \qquad k \geq 0, \tag{P}$$

until some termination criterion is satisfied. Although numerous, possibly faster, methods have been proposed since 1998 [4, 31, 32], the power method retains many advantages:

1. It is simple to implement, especially in a parallel computing environment [17].
2. It requires a minimum of storage.
3. It has a robust and predictable convergence behavior that is insensitive to changes in the matrix. The convergence rate depends only on α, and is not sensitive to the underlying Web graph represented by S, the personalization vector v, the dangling node vector, or the starting vector $x^{(0)}$ [17].
4. It is numerically stable. All operations are numerically well conditioned. If $1 - \alpha$ is precomputed, then no subtractions are necessary to compute the iterates of (P). There is no danger of overflow since, in exact arithmetic, $\|x^{(k)}\|_1 = 1$.

The power method (P) as well as most iterative methods for computing PageRank compare successive iterates, based on their geometric distance or ranking distance. Once such a distance is small enough, the most recent iterate is judged to be a sufficiently good approximation to π. But the question is: Is the induced ranking of the iterate correct?

Overview. In section 2 we answer the previous question with "no." In section 3 we present the main idea of our paper, a ranking criterion for the elements of an iterate $x^{(k)}$:

If $x_i^{(k)} > x_j^{(k)} + \beta$, then $\pi_i > \pi_j$.

Here β is an upper bound for the error $\|x^{(k)} - \pi\|_1$. We show how this criterion can be used for exact, top k, and bucket ranking. In section 4 we derive several classes of computationally efficient bounds β based on geometric distances between iterates. In section 5 we present a careful implementation of the power method and derive roundoff error bounds for the iterates $x^{(k)}$. Numerical experiments in section 6 demonstrate that our criterion can identify the ranking of the top PageRank scores. We conclude with a discussion of how to solve extremely large problems in section 7.

Notation. All matrices are $n \times n$ matrices, and all vectors are $n \times 1$ column vectors. The $n \times n$ identity matrix is I, with ith column e_i. The transpose of a vector v is v^T, the elements of v are v_i, and inequalities like $v \geq 0$ and $|v| \geq 0$ are to be interpreted componentwise. The one norm $\|\cdot\|_1$ is the maximal column sum, and the infinity norm $\|\cdot\|_\infty$ the maximal row sum. In particular, if x is a column vector, then

$$\|x\|_1 = \sum_i |x_i| = \|x^T\|_\infty, \qquad \|x\|_\infty = \max_i |x_i| = \|x^T\|_1.$$

2. Existing termination criteria do not produce correct rankings. The two most frequently mentioned termination criteria for PageRank computation are based on geometric distance and ranking distance [6, section 4.2.1].

Geometric distance. The traditional convergence criterion terminates the power method once some norm of the residual is sufficiently small. Since the power method iteration (P) contains no normalization, the residual equals the difference between successive iterates, $[x^{(k)}]^T - [x^{(k)}]^T G = [x^{(k)} - x^{(k+1)}]^T$. The power method (P) is terminated once $\|x^{(k)} - x^{(k+1)}\|$ is sufficiently small in the one, two, or infinity norms. The residual norm of (P) can be interpreted as a distance between two vectors, and has thus been classified as a *geometric distance* [6, section 4.2.1], in contrast to the *ranking distance* below.

Ranking distance. We specify the position of an element in an ordered list through its *ordinal rank*, which is defined below first for a single element and then for a whole vector (the ordinal rank bears no relation to the numerical rank).

DEFINITION 2.1 (ordinal rank). *Let* $x = \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix}^T$ *be a real vector, and* σ *a permutation that orders the elements of* x *in decreasing order,* $x_{\sigma(1)} \geq \dots \geq x_{\sigma(n)}$. *Then the ordinal rank of an individual element is* $\mathrm{Orank}(x_i) \equiv \sigma(i)$, $1 \leq i \leq n$, *and the ordinal rank of the whole vector is* $\mathrm{Orank}(x) \equiv \begin{pmatrix} \mathrm{Orank}(x_1) & \dots & \mathrm{Orank}(x_n) \end{pmatrix}$.

If the elements of x are distinct, and if $x_j = \max_i x_i$, then $\mathrm{Orank}(x_j) = 1$; and if $x_j = \min_i x_i$, then $\mathrm{Orank}(x_j) = n$. If x contains identical elements, then the ordinal ranking is not unique, because the permutation σ in Definition 2.1 is not unique and is not required to preserve the relative order of the elements (i.e., σ does not have to be stable in the sense of sorting). In contrast to other ranking schemes, no two items receive the same ordinal rank, even when they are equal. The concept of ordinal rank leads to the particular ranking distance below, which is known as *Kendall's* τ *distance* [6, section 4.2.2] [27, section 1.13].

DEFINITION 2.2 (ranking distance). *The ranking distance between real vectors* x *and* y, *each with distinct elements, is* $\sum_i \sum_j \delta_{xy}(i,j)$, *where*

$$\delta_{xy}(i,j) \equiv \begin{cases} 1 & \mathrm{Orank}(x_i) < \mathrm{Orank}(x_j) \text{ and } \mathrm{Orank}(y_i) > \mathrm{Orank}(y_j) \\ 0 & \text{otherwise} \end{cases}.$$

Ranking distances have been used as termination criteria in iterative methods for computing PageRank [3, 10, 25, 28, 38, 39, 46]. Because the matrix G is large, computing the ranking distance for entire vectors is too expensive. To reduce computation time, one can focus on the top k rankings [15, 25, 26]. Experiments in [26] suggest that termination based on the top k rankings tends to produce rankings that resemble those produced by a termination criterion based on the one norm of the residual.

Incorrect ranking. Simple examples, such as those described in [47, section 4.2] for the directed ring[1] graph with n vertices, illustrate that geometric and ranking distances between successive iterates of the power method (P) can fail to produce correct rankings. In addition, the examples demonstrate that (1) correct ranking can be achieved in some iteration and destroyed in the next, (2) a small residual norm does not guarantee correct ranking, (3) zero ranking distance between successive iterates does not guarantee correct ranking, and (4) successive iterates can be correctly ranked before the residual norm is small.

3. Ranking. Since geometric and ranking distances between successive iterates of the power method (P) do not ensure correct ranking, we consider instead the

[1] In a directed ring graph with n vertices, vertex i links to vertex $i+1$, $1 \leq i \leq n-1$, and vertex n links back to vertex 1.

ranking distance between an iterate $x^{(k)}$ and the desired PageRank vector π. We obtain information about the ranking distance from the geometric distance, and show how the resulting ranking criterion can be used to perform exact, top k, and bucket ranking.

We use a consequence of an inequality from [21, Corollary 2.4(a)] which relates the one norm and infinity norm of a vector whose elements sum to zero: If y is a column vector with $y^T\mathbf{1} = 0$, then

$$\|y\|_\infty \leq \|y\|_1/2. \tag{3.1}$$

The main idea of our paper is to gather information about relative ranking based on an approach by Kirkland [29, Corollaries 3.9-3.12], which we present below in a more general context.

THEOREM 3.1 (ranking criterion). *Let $x \geq 0$ with $\|x\|_1 = 1$ be an approximation to π in* (G)*, and $\beta \geq \|x - \pi\|_1$. If $x_i > x_j + \beta$, then $\pi_i > \pi_j$.*

Proof. If $x = \pi$, then $\beta = 0$ and the result holds trivially. Assuming $\beta > 0$ gives $(x_i - \pi_i) - (x_j - \pi_j) \leq |x_i - \pi_i| + |x_j - \pi_j| \leq 2\|x - \pi\|_\infty$. Since $(x - \pi)^T\mathbf{1} = 0$, (3.1) implies $\|x - \pi\|_\infty \leq \|x - \pi\|_1/2$. Hence $(x_i - \pi_i) - (x_j - \pi_j) \leq \|x - \pi\|_1 \leq \beta$, and $x_i - (x_j + \beta) \leq \pi_i - \pi_j$. Therefore, $0 < x_i - (x_j + \beta)$ implies $0 < \pi_i - \pi_j$. ∎

Consequently, if two elements of $x^{(k)}$ are well-separated (compared to the geometric distance between $x^{(k)}$ and π), then we can say something about the relative rankings of the corresponding PageRank scores. Because the ranking criterion in Theorem 3.1 applies only to well-separated elements, it can, in general, determine only a partial ranking of the PageRank scores.

Kirkland [29, section 3] expresses the quantity β in Theorem 3.1 as a function of the lengths of shortest cycles on which vertices i and j are situated in the graph of S. However, it is not clear how to efficiently compute shortest cycle lengths for all vertices.

Exact, top k, and bucket ranking. The bucket ranking criteria below are motivated by Google's Toolbar PageRank scores, which are integers from 0 (low) to 10 (high). Our ranking criteria determine a topological (or partial) order for the PageRank scores.

Let $x \geq 0$ with $\|x\|_1 = 1$ be an approximation to π in (G) and $\beta \geq \|x - \pi\|_1$. Let Q be a permutation that orders the elements of x in decreasing magnitude, i.e.,

$$\tilde{x} \equiv Qx = \begin{pmatrix} \tilde{x}_1 & \dots & \tilde{x}_n \end{pmatrix}^T, \qquad \tilde{x}_1 \geq \dots \geq \tilde{x}_n, \qquad \tilde{\pi} \equiv Q\pi = \begin{pmatrix} \tilde{\pi}_1 & \dots & \tilde{\pi}_n \end{pmatrix}^T.$$

In contrast to the elements of $\tilde{x}$, those of $\tilde{\pi}$ are, in general, not ordered in decreasing magnitude.

First we show that if element k of $\tilde{x}$ is well separated from element $k + 1$, then elements $1, \dots, k$ approximate the top k PageRank scores.

LEMMA 3.2 (top k). *If $\tilde{x}_k > \tilde{x}_{k+1} + \beta$, then* $\text{Orank}(\tilde{\pi}_i) \leq k$ *for $1 \leq i \leq k$, and* $\text{Orank}(\tilde{\pi}_j) \geq k + 1$ *for $k + 1 \leq j \leq n$.*

Proof. From $\tilde{x}_k > \tilde{x}_{k+1} + \beta$ follows $\tilde{\pi}_k > \tilde{\pi}_{k+1}$, according to Theorem 3.1. The descending ordering implies $\tilde{x}_k > \tilde{x}_{k+1} + \beta \geq \dots \geq \tilde{x}_n + \beta$, so that $\tilde{\pi}_k > \tilde{\pi}_{k+1}, \dots, \tilde{\pi}_n$. The descending ordering also implies $\tilde{x}_1 \geq \dots \geq \tilde{x}_k > \tilde{x}_{k+1} + \beta$, so that $\tilde{\pi}_1, \dots, \tilde{\pi}_k > \tilde{\pi}_{k+1}$. Combining the two sets of inequalities yields $\tilde{\pi}_1, \dots, \tilde{\pi}_k > \tilde{\pi}_{k+1}, \dots, \tilde{\pi}_n$. Therefore $\text{Orank}(\tilde{\pi}_i) \leq k$ for $1 \leq i \leq k$, and $\text{Orank}(\tilde{\pi}_j) \geq k + 1$ for $k + 1 \leq j \leq n$. ∎

Now we present a criterion for finding the "exact" rank (here "exact" does not refer to finite precision accuracy but to the fact that we can assign a number, rather

than an interval, to the rank). If element $k+1$ of $\tilde{x}$ is well separated from elements k and $k+2$, then element $k+1$ of $\tilde{x}$ approximates the $(k+1)$st PageRank score.

LEMMA 3.3 (exact rank). *If* $\tilde{x}_k > \tilde{x}_{k+1} + \beta$ *and* $\tilde{x}_{k+1} > \tilde{x}_{k+2} + \beta$, *then* $\mathrm{Orank}(\tilde{\pi}_{k+1}) = k+1$.

Proof. This follows from Lemma 3.2. Condition $\tilde{x}_k > \tilde{x}_{k+1}+\beta$ implies $\tilde{\pi}_1, \ldots, \tilde{\pi}_k > \tilde{\pi}_{k+1}$; hence $\mathrm{Orank}(\tilde{\pi}_{k+1}) \geq k+1$. Condition $\tilde{x}_{k+1} > \tilde{x}_{k+2} + \beta$ implies $\tilde{\pi}_{k+1} > \tilde{\pi}_{k+2}, \ldots, \tilde{\pi}_n$; hence $\mathrm{Orank}(\tilde{\pi}_{k+1}) \leq k+1$. ∎

Often it is not possible to determine the exact PageRank, but we can still try to assign PageRank scores to a "bucket". This is done in the next lemma, which represents an extension of Lemma 3.3 to intervals.

LEMMA 3.4 (bucket). *If* $\tilde{x}_k > \tilde{x}_{k+i} + \beta$ *and* $\tilde{x}_{k+i} > \tilde{x}_{k+i+j} + \beta$ *for* $i, j \geq 1$ *then* $k+1 \leq \mathrm{Orank}(\tilde{\pi}_{k+i}) \leq k+i+j-1$.

Proof. This follows from Lemma 3.2. The condition $\tilde{x}_k > \tilde{x}_{k+i} + \beta$ implies $\tilde{\pi}_1, \ldots, \tilde{\pi}_k > \tilde{\pi}_{k+i}$, so that $\mathrm{Orank}(\tilde{\pi}_{k+i}) \geq k+1$. The condition $\tilde{x}_{k+i} > \tilde{x}_{k+i+j} + \beta$ implies $\tilde{\pi}_{k+i} > \tilde{\pi}_{k+i+j}, \ldots, \tilde{\pi}_n$ so that $\mathrm{Orank}(\tilde{\pi}_{k+i}) \leq k+i+j-1$. ∎

Lemma 3.4 assigns PageRank score $\tilde{x}_{k+i}$ to a bucket that represents ranks $k+1, \ldots, k+i+j-1$. If $i = j = 1$ then the bucket consists of a single number and Lemma 3.4 reduces to Lemma 3.3. The top k ranking in Lemma 3.2 is a special case of bucket ranking with two buckets: One for ranks $1, \ldots, k$ and another one for ranks $k+1, \ldots, n$. In contrast to bucket sorting the buckets are not specified beforehand; they emerge during the execution of the power method (P).

4. Error bounds for the power method. We present four classes of computable bounds for the error $\|x^{(k)} - \pi\|_1$ and for the quantity β used in the ranking criteria in section 3. The four classes consist of the following types of bounds: simple (section 4.1), backward looking (section 4.2), forward looking (section 4.3), and two-level forward looking (section 4.4). We use the following facts for stochastic matrices S: $\|S^i\|_\infty = 1$, $i \geq 0$, and

$$\|S^i(I - \alpha^j S^j)^{-1}\|_\infty = \frac{1}{1-\alpha^j}, \qquad i \geq 0, \quad j \geq 1. \tag{4.1}$$

The last expression follows from the fact that $(I - \alpha^j S^j)^{-1}$ is nonnegative, and $\|(I - \alpha^j S^j)^{-1}\|_\infty = (1-\alpha^j)^{-1}$ [32, section 7.1].

First we justify why the ranking of the iterates in (P) can converge under a criterion like the one in Theorem 3.1. The inequalities below relate two corresponding components of two different iterates and show that the distance between the components changes less and less as the iterations progress.

THEOREM 4.1 (component distances stabilize).

$$|x_i^{(k-1)} - x_j^{(k-1)}| - \alpha^{k-1}\gamma \leq |x_i^{(k)} - x_j^{(k)}| \leq |x_i^{(k-1)} - x_j^{(k-1)}| + \alpha^{k-1}\gamma, \qquad k \geq 1,$$

where $\gamma \equiv 2\|x^{(1)} - x^{(0)}\|_1$.

Proof. As in [7, Property 7] one shows $[x^{(k)}]^T = \alpha^k [x^{(0)}]^T S^k + (1-\alpha)\sum_{l=0}^{k-1} \alpha^l v^T S^l$. Hence the difference between two components equals

$$\begin{aligned} x_i^{(k)} - x_j^{(k)} &= \alpha^k \left[x^{(0)}\right]^T S^k (e_i - e_j) + (1-\alpha)\sum_{l=0}^{k-1} \alpha^l v^T S^l (e_i - e_j) \\ &= \left(x_i^{(k-1)} - x_j^{(k-1)}\right) + \alpha^{k-1}\left[x^{(1)} - x^{(0)}\right]^T S^{k-1}(e_i - e_j). \end{aligned}$$

The Hölder inequality and (4.1) imply

$$\left|[x^{(1)} - x^{(0)}]^T S^{k-1}(e_i - e_j)\right| \leq \|x^{(1)} - x^{(0)}\|_1 \|S^{k-1}(e_i - e_j)\|_\infty \leq 2\|x^{(1)} - x^{(0)}\|_1. \quad \square$$

The idea behind the bounds for $\|x^{(k)} - \pi\|_1$ is to extend the residual, which is a difference of successive iterates $[x^{(k)}]^T - [x^{(k)}]^T G = [x^{(k)} - x^{(k+1)}]^T$, to differences between nonsuccessive iterates $[x^{(k-j)} - x^{(k)}]^T$. The derivations in subsequent sections are based on the following recursions.

LEMMA 4.2 (recursions).

$$\begin{aligned} \text{Error :} \quad & [x^{(k+1)} - \pi]^T = \alpha[x^{(k)} - \pi]^T S, \qquad k \geq 0 \\ \text{Iterate difference :} \quad & [x^{(k-j+1)} - x^{(k+1)}]^T = \alpha[x^{(k-j)} - x^{(k)}]^T S, \qquad 1 \leq j \leq k. \end{aligned}$$

Proof. The recursions follow from $[x^{(k+1)}]^T = [x^{(k)}]^T G$, $G = \alpha S + (1-\alpha)\mathbf{1}v^T$, and $[x^{(k)} - \pi]^T \mathbf{1} = 0$.

Note that the statements also follow from the properties of splitting methods [43, section 3.6]. This is because π is the solution of the linear system $\pi^T(I - \alpha S) = (1-\alpha)v^T$, and the power method $[x^{(k+1)}]^T = [x^{(k)}]^T G$ is mathematically equivalent to a splitting method $[x^{(k+1)}]^T M = [x^{(k)}]^T N + (1-\alpha)v^T$, where $M = I$ and $N = \alpha S$. $\square$

The second recursion in Lemma 4.2 is an extension of the one derived for $j = 1$ in [29, Proof of Corollary 3.11].

4.1. Simple bound. The recursions in Lemma 4.2 lead immediately to a simple normwise error bound that depends only on α and k.

THEOREM 4.3 (simple bound). $\|x^{(k)} - \pi\|_1 \leq \alpha^k \|x^{(0)} - \pi\|_1 \leq 2\alpha^k$, $k \geq 0$.

Proof. Lemma 4.2 implies $[x^{(k)} - \pi]^T = \alpha^k [x^{(0)} - \pi]^T S^k$, and (4.1) implies

$$\|x^{(k)} - \pi\|_1 = \|[x^{(k)} - \pi]^T\|_\infty \leq \alpha^k \|[x^{(0)} - \pi]^T\|_\infty \|S^k\|_\infty = \alpha^k \|x^{(0)} - \pi\|_1.$$

The second upper bound follows from the triangle inequality and $\|x^{(0)}\|_\infty = \|\pi\|_\infty = 1$. $\square$

The bounds in Theorem 4.3 first appeared in [5, Theorem 5.1] and [24, section 4]. In the special case $x^{(0)} = v$, when the starting vector equals the personalization vector, the power of α can be increased by one, $\|x^{(k)} - \pi\|_1 \leq \alpha^{k+1}\|v - S^T\pi\|_1 \leq 2\alpha^{k+1}$; see also [7, Property 9]. The simple bound in Theorem 4.3 can be used as a ranking criterion in the power method (P) as follows.

COROLLARY 4.4 (ranking with the simple bound). *If* $x_i^{(k)} > x_j^{(k)} + 2\alpha^k$, $k \geq 1$, *then* $\pi_i > \pi_j$.

Proof. Follows from Theorem 4.3 and setting $\beta = 2\alpha^k$ in Theorem 3.1. $\square$

4.2. Backward-looking bounds. Backward-looking bounds are constructed from previous iterates.

THEOREM 4.5 (backward-looking bounds). $\|x^{(k)} - \pi\|_1 \leq \frac{\alpha^j}{1-\alpha^j}\|x^{(k-j)} - x^{(k)}\|_1$, $1 \leq j \leq k$.

Proof. Lemma 4.2 implies

$$\begin{aligned} \alpha^j [x^{(k-j)} - x^{(k)}]^T S^j &= [x^{(k)} - x^{(k+j)}]^T = [x^{(k)} - \pi]^T - [x^{(k+j)} - \pi]^T \\ &= [x^{(k)} - \pi]^T - \alpha^j [x^{(k)} - \pi]^T S^j = [x^{(k)} - \pi]^T (I - \alpha^j S^j). \end{aligned}$$

Hence $\left[x^{(k)} - \pi\right]^T = \alpha^j \left[x^{(k-j)} - x^{(k)}\right]^T S^j (I - \alpha^j S^j)^{-1}$. Take norms

$$\|x^{(k)} - \pi\|_1 = \|[x^{(k)} - \pi]^T\|_\infty \leq \alpha^j \|[x^{(k-j)} - x^{(k)}]^T\|_\infty \|S^j (I - \alpha^j S^j)^{-1}\|_\infty$$

and use (4.1). $\square$

The bound for $j = 1$ in Theorem 4.5 was already derived in [29, (3.1)] for general approximations, not limited to just power method iterates. Note that for a fixed step length j, a full backward look is not possible in early iterations as long as $k < j$. Experiments indicate that no bound is always the tightest. We can distinguish two types of backward-looking bounds: Those where j is fixed and those where j is a function of k. Among the fixed step bounds, the $j = 1$ bound is preferable for several reasons: It tends to be competitive in the long-term since $\|x^{(k)} - \pi\|_1 \leq \alpha\|x^{(k-1)} - \pi\|_1 \leq \alpha^2\|x^{(k-2)} - \pi\|_1 \leq \ldots$. It takes effect immediately starting with iteration 1 – in contrast to other bounds which require a startup of j iterations before the full backward look is possible. It performs well in our experiments in later iterations. At last, it keeps storage requirements low, because bounds with fixed j need to store j or more vectors. Among the bounds where j is a function of k, the simplest one is $j = k$,

$$\|x^{(k)} - \pi\|_1 \leq \frac{\alpha^k}{1-\alpha^k}\|x^{(0)} - x^{(k)}\|_1.$$

This bound has low storage requirements as well, it is effective immediately, starting with iteration 1, and in our experiments it tends to do well in early iterations. However, the $j = k$ bound does not do as well in later iterations. This is because it depends on the initial error $\|x^{(0)} - \pi\|_1$ which remains constant throughout the iterations, while for bounds with a fixed step length j, the error $\|x^{(k-j)} - \pi\|_1$ approaches zero as k increases. Applied to the power method (P), the backward-looking bounds in Theorem 4.5 can be used for ranking as follows.

COROLLARY 4.6 (ranking with backward-looking bounds). *If* $x_i^{(k)} > x_j^{(k)} + \frac{\alpha^l}{1-\alpha^l}\|x^{(k-l)} - x^{(k)}\|_1$, $1 \leq l \leq k$, *then* $\pi_i > \pi_j$.

4.3. Forward-looking bounds. Forward-looking bounds are constructed from future iterates. The derivation is similar to the one in Theorem 4.5.

THEOREM 4.7 (forward-looking bounds). $\|x^{(k)} - \pi\|_1 \leq \frac{\|x^{(k+j)} - x^{(k)}\|_1}{1-\alpha^j}$, $k \geq 0$, $j \geq 1$.

The forward-looking bound for $j = 1$ was derived in [7, Property 12]. Looking farther ahead can lead to better estimates for the error at the current iteration k. This is to be expected because future iterates can be more accurate. Comparing the bounds in Theorem 4.7 for $j = 1$ and $j > 1$ shows that looking several steps ahead can result in tighter bounds than just looking a single step ahead,

$$\frac{\|x^{(k+j)} - x^{(k)}\|_1}{1-\alpha^j} \leq \frac{\|x^{(k+1)} - x^{(k)}\|_1}{1-\alpha}, \qquad k \geq 0, \quad j \geq 1.$$

Applied to the power method (P), the forward-looking bounds in Theorem 4.7 can be used for ranking as follows.

COROLLARY 4.8 (ranking with forward-looking bounds). *If* $x_i^{(k)} > x_j^{(k)} + \frac{\|x^{(k+l)} - x^{(k)}\|_1}{1-\alpha^l}$, $k \geq 0$, $l \geq 1$, *then* $\pi_i > \pi_j$.

4.4. Two-level, forward-looking bounds. Another type of forward bound looks forward in two stages.

THEOREM 4.9 (two-level, forward-looking bounds).

$$\|x^{(k)} - \pi\|_1 \leq \|x^{(k+j)} - x^{(k)}\|_1 + \frac{\|x^{(k+j+i)} - x^{(k+j)}\|_1}{1-\alpha^i}, \qquad k \geq 0, \quad j, i \geq 1.$$

COROLLARY 4.10 (ranking with two-level, forward-looking bounds). *If*

$$x_i^{(k)} > x_j^{(k)} + \|x^{(k+l)} - x^{(k)}\|_1 + \frac{\|x^{(k+l+h)} - x^{(k+l)}\|_1}{1-\alpha^h}, \qquad k \geq 0, \quad l,h \geq 1,$$

then $\pi_i > \pi_j$.

5. Finite precision computation. We present error bounds for perturbed power method iterates in section 5.1, a floating point implementation of the power method in section 5.2, and bounds for ranking in floating point arithmetic in section 5.3.

5.1. Perturbation bounds. In a finite precision context, the ranking criterion in Theorem 3.1 must be applied to perturbed power method iterates $\hat{x}^{(k)}$. That is, if $\hat{x}_i^{(k)} > \hat{x}_j^{(k)} + \|\hat{x}^{(k)} - \pi\|_1$, then $\pi_i > \pi_j$. We assume that the perturbed iterates are nonnegative, have unit norm, and incur an error during each iteration. The error bounds for $\|\hat{x}^{(k)} - \pi\|_1$ below are simple and easy to compute.

THEOREM 5.1 (finite precision error bounds). *Let* $\hat{x}^{(0)} = x^{(0)}$, *and* $[\hat{x}^{(k+1)}]^T = [\hat{x}^{(k)}]^T G + g_k^T$, $k \geq 0$, *be such that* $\hat{x}^{(k)} \geq 0$ *and* $\|\hat{x}^{(k)}\|_1 = 1$, $k \geq 0$.

1. *Simple bound:*

$$\|\hat{x}^{(k)} - \pi\|_1 \leq 2\alpha^k + \frac{1-\alpha^k}{1-\alpha} \max_{0\leq i\leq k-1} \|g_{k-i}\|_1, \qquad k \geq 1.$$

2. *Backward-looking bounds:*

$$\|\hat{x}^{(k)} - \pi\|_1 \leq \frac{\alpha^j}{1-\alpha^j}\|\hat{x}^{(k-j)} - \hat{x}^{(k)}\|_1 + \frac{1-\alpha^j}{1-\alpha} \max_{0\leq i\leq j-1} \|g_{k-i}\|_1, \qquad 0 \leq j < k.$$

3. *Forward-looking bounds:*

$$\|\hat{x}^{(k)} - \pi\|_1 \leq \frac{\|\hat{x}^{(k+j)} - \hat{x}^{(k)}\|_1}{1-\alpha^j} + \frac{1-\alpha^j}{1-\alpha} \max_{1\leq i\leq j} \|g_{k+i}\|_1, \qquad k \geq 0, \quad j \geq 1.$$

4. *Two-level, forward-looking bounds:*

$$\|\hat{x}^{(k)} - \pi\|_1 \leq \|\hat{x}^{(k+j)} - \hat{x}^{(k)}\|_1 + \frac{\|\hat{x}^{(k+j+i)} - \hat{x}^{(k+j)}\|_1}{1-\alpha^i} + \frac{1-\alpha^i}{1-\alpha} \max_{1\leq l\leq i} \|g_{k+j+l}\|_1,$$

where $k \geq 0$, $j,i \geq 1$.

Proof. First we derive an expression for the absolute error in the perturbed iterates. From $\hat{x}^{(k)} \geq 0$, $x^{(k)} \geq 0$, $\|\hat{x}^{(k)}\|_1 = \|x^{(k)}\|_1 = 1$ follows $g_k^T\mathbf{1} = f_k^T\mathbf{1} = 0$. With $f_0 \equiv 0$ this implies

$$\hat{x}^{(k)} = x^{(k)} + f_k, \qquad f_k^T \equiv \alpha^j f_{k-j}^T S^j + \sum_{i=0}^{j-1} \alpha^i g_{k-i}^T S^i, \qquad 1 \leq j \leq k. \tag{5.1}$$

We use (5.1) to derive each of the four finite precision bounds.

1. Simple bound: This follows from $[\hat{x}^{(k)} - \pi]^T = [x^{(k)} - \pi]^T + f_k^T$ and (5.1).
2. Backward-looking bounds: As in the proof of Theorem 4.5 one shows

$$[\hat{x}^{(k)} - \pi]^T(I - \alpha^j S^j) = \alpha^j[\hat{x}^{(k-j)} - \hat{x}^{(k)}]^T S^j + f_k^T - \alpha^j f_{k-j}^T S^j.$$

Then (5.1) implies $f_k^T - \alpha^j f_{k-j}^T S^j = \sum_{i=0}^{j-1} \alpha^i g_{k-i}^T S^i$.

3. Forward-looking bounds: As in the proof of Theorem 4.7 one shows

$$[\hat{x}^{(k)} - \pi]^T (I - \alpha^j S^j) = [\hat{x}^{(k)} - \hat{x}^{(k+j)}]^T + f_{k+j}^T - \alpha^j f_k^T S^j.$$

From (5.1) follows $f_{k+j}^T - \alpha^j f_k^T S^j = \sum_{i=0}^{j-1} \alpha^i g_{k+j-i}^T S^i$.

4. Two-level, forward-looking bounds: As in the proof of Theorem 4.9, one shows that $[\pi - \hat{x}^{(k)}]^T$ is equal to

$$[\hat{x}^{(k+j)} - \hat{x}^{(k)}]^T - [\hat{x}^{(k+j)} - \hat{x}^{(k+j+i)}]^T (I - \alpha^i S^i)^{-1} - f_{k+j}^T - [f_{k+j+i} - f_{k+j}]^T (I - \alpha^i S^i)^{-1}.$$

Hence (5.1) implies for the error term

$$-f_{k+j}^T - [f_{k+j+i} - f_{k+j}]^T (I - \alpha^i S^i)^{-1} = -\sum_{l=0}^{i-1} \alpha^l g_{k+j+i-l}^T S^l (I - \alpha^i S^i)^{-1}. \qquad \square$$

The term g_k takes care of finite precision errors incurred in iteration k, including those from matrix vector multiplication, as well as explicit normalization of the iterates if necessary. Theorem 5.1 shows that the bounds are affected only by the error in a single iteration, and do not suffer from accumulation of errors.

5.2. Power method implementation. We discuss the implementation of the power method in floating point arithmetic.

As already mentioned in section 1, the matrix S is derived from the webgraph and zero rows corresponding to dangling nodes (i.e., web pages without outlinks) are modified to ensure that S is stochastic. Computationally, though, it is more efficient to keep the webgraph part separated from the dangling node fix so that one can take advantage of the latter's low rank [32, section 8.1]. It turns out that this separation also limits accumulation of roundoff error in a matrix vector multiplication with S. Therefore it is necessary to discuss the construction of S in more detail.

The web graph is represented by a $n \times n$ *substochastic* matrix H. That is, the elements of H are nonnegative, and each row is either zero, or else its elements sum to one. The zero rows correspond to dangling nodes, which are web pages without outlinks. To obtain the stochastic matrix S, one can replace each zero row by the same dangling node vector w^T, where w is a column vector with $w \geq 0$ and $\|w\|_1 = 1$. The resulting Google matrix is $G = \alpha S + (1 - \alpha)\mathbf{1}v^T$, where $S = H + dw^T$ and d is a column vector of zeros and ones. An element of d is equal to 1 if the corresponding row in H is zero; otherwise this element of d is equal to zero. The following floating point implementation of the power method (P) exploits the fact that dw^T and $\mathbf{1}v^T$ have rank one.

Floating point implementation of (P). Let $\hat{x}^{(0)} \geq 0$ with $\|\hat{x}^{(0)}\|_1 = 1$, and $\alpha_1 \equiv 1 - \alpha$. Repeat

$$\text{(FP)} \qquad [y^{(k+1)}]^T := \text{fl}\left(\alpha([\hat{x}^{(k)}]^T H + ([\hat{x}^{(k)}]^T d)w^T) + \alpha_1 v^T\right)$$
$$\hat{x}^{(k+1)} := \text{fl}\left(y^{(k+1)}/\|y^{(k+1)}\|_1\right)$$

until some termination criterion is satisfied.

The explicit normalization of the iterates in (FP) is necessary to ensure that iterate norms remain close to unity in a finite precision environment. Figure 5.1 illustrates why this is necessary. The norms of the unnormalized iterates $y^{(k)}$ deviate much further from 1 than the norms of the normalized iterates $\hat{x}^{(k)}$. The ratios

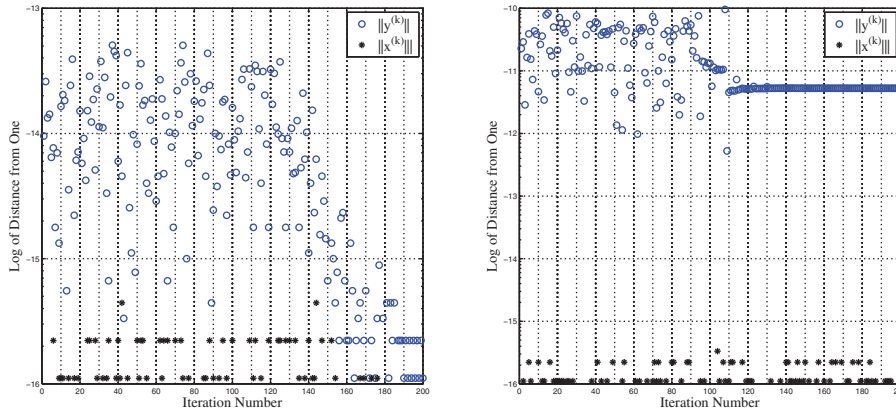


FIG. 5.1. $|1 - \|\hat{x}^{(k)}\|_1|$ *and* $|1 - \|y^{(k)}\|_1|$ *for iterates of the Power Method* (FP) *applied to the matrices* G_S *(left) and* G_L *(right) in section* 6.

$|1 - \|y^{(k)}\|_1|$ can approach 10^{-13} for the small matrix G_S and hover around 10^{-11} for the larger matrix G_L. The analysis in the proof of Theorem 5.2 explains this: In IEEE double precision the unit roundoff is $\epsilon \approx 10^{-16}$, so that the error in the matrix vector multiplication $[\hat{x}^{(k)}]^T G_L$ is about $m\epsilon \approx 4 \cdot 10^{-11}$, where $m = 168,685$ is the maximal number of nonzero elements in any column of H_L. For the smaller matrix G_S, the computation of $x^T d$ dominates, leading to an error of $\|d\|_1 \epsilon \approx 7 \cdot 10^{-12}$, since d has 2,861 elements equal to one. In contrast, the norms of the normalized iterates are almost perfect. In all iterations k the deviation $|\|\hat{x}^{(k)}\|_1 - 1|$ is either ϵ, $\epsilon/2$, or 0. Note that Figure 5.1 shows merely the effect of a single missing normalization; the accumulated damage from failing to normalize over many iterations is much worse.

5.3. Floating point bounds. We bound the roundoff error g_k incurred in iteration k of the power method (FP). Existing roundoff error bounds for the power method and stationary iterative methods [22, sections 17, 18], [42] do not seem to be applicable here, because they require knowledge of the condition number of a diagonalizing transformation, or assume that the spectral radius of the iteration matrix is strictly less than one.

We assume the standard model for the elementary floating point arithmetic operations with unit roundoff ϵ [22, section 2.2]. If a and b are floating point numbers, then

$$\mathrm{fl}(a \text{ op } b) = (a \text{ op } b)(1 + \delta), \qquad |\delta| \leq \epsilon, \quad \text{op} = +, -, *, /.$$

We exploit the fact that the iterations in (FP) execute no subtractions, and that all operations are well conditioned. The norms are computed by *compensated summation* [22, section 4.3], [35] so that the dominant part of the roundoff error $\|g_k\|_1$ does not depend on the matrix dimension n, but only on the sparsity of the matrix. The analysis below holds for matrices of order $n < 10^{14}$, in IEEE double precision with unit roundoff $\epsilon \equiv 10^{-16}$.

THEOREM 5.2 (floating point bounds). *Assume that* $n\epsilon < .01$, *and*

1. *The scalars* α *and* $\alpha_1 \equiv 1 - \alpha$, *and the elements of* H, v, w, *and* $\hat{x}^{(0)} = x^{(0)}$ *are floating point numbers.*
2. *The iterates* $\hat{x}^{(k)}$ *are computed according to* (FP).
3. *The norms* $\|y^{(k+1)}\|_1$ *in* (FP) *are computed by compensated summation.*

4. *H has at most m nonzeros per column, and $\|d\|_1$ zero rows.*
Then

$$\|g_{k+1}\|_1 \le \frac{2\epsilon(3.03 + c\alpha M)}{1 - \epsilon(3.03 + c\alpha M)} + \mathcal{O}(n\epsilon^2), \qquad k \ge 0$$

where $M \equiv \max\{m, \|d\|_1 + 1\}$, and $c \equiv 1.01(1 + 3.03\epsilon)$.

Proof. Abbreviate $x \equiv \hat{x}^{(k)}$, $y \equiv y^{(k+1)}$, and $\delta \equiv \|d\|_1 + 1$.

1. Since H has at most m nonzero elements per column, the elements of S and x are nonnegative, and $m\epsilon \le .01$ we obtain with [22, (3.2) and Lemma 3.4] $\mathrm{fl}(x^T H) = x^T H + h_1^T$, where $|h_1^T| \le 1.01 m\epsilon\,(x^T H)$.

2. Similarly, since α and the elements of d and x are nonnegative, we obtain for the second summand in $\alpha x^T S$, $\mathrm{fl}(\alpha x^T d) = \alpha x^T d + h_2$, where $|h_2| \le 1.01\delta\epsilon\,(\alpha x^T d)$.

3. For the computation of $y^T = \mathrm{fl}(x^T G)$ abbreviate $z^T \equiv \alpha\,\mathrm{fl}(x^T H) + \mathrm{fl}(\alpha x^T d) w^T + \alpha_1 v^T$ so that $y^T = \mathrm{fl}(z^T) = z^T + h_3^T$, where $|h_3^T| \le 3.03\epsilon\, z^T$. From $z^T = x^T G + \alpha h_1^T + h_2 w^T$ follows

$$\begin{aligned} |h_3^T| &\le 3.03\epsilon\, \left(x^T G + 1.01 m\epsilon\,(x^T H) + 1.01\delta\epsilon\,(\alpha x^T d w^T)\right) \\ &\le 3.03\epsilon\, \left(x^T G + 1.01 M\epsilon\,(\alpha x^T S)\right). \end{aligned}$$

In order to express y^T in terms of $x^T G$, write $y^T = z^T + h_3^T = x^T G + \alpha h_1^T + h_2 w^T + h_3^T$. Then $y^T = x^T G + h_4^T$, where $h_4^T \equiv \alpha h_1^T + h_2 w^T + h_3^T$. Hence

$$\begin{aligned} |h_4^T| &\le 1.01 m\epsilon\,(\alpha x^T H) + 1.01\delta\epsilon\,(\alpha x^T d w^T) + 3.03\epsilon\, \left(x^T G + 1.01 M\epsilon\,(\alpha x^T S)\right) \\ &\le 1.01 M\epsilon\,(\alpha x^T S) + 3.03\epsilon\, \left(x^T G + 1.01 M\epsilon\,(\alpha x^T S)\right) \\ &\le \epsilon\left(3.03\, x^T G + 1.01 M(1 + 3.03\epsilon)\,(\alpha x^T S)\right). \end{aligned}$$

4. Now comes the computation of $\|y\|_1$. Since $n\epsilon < .01$, and the additions in $\|y\|_1$ involve only nonnegative numbers, a compensated summation gives [22, (4.8)] $\eta \equiv \mathrm{fl}(\|y\|_1) = \|y\|_1(1 + h_5)$, where $|h_5| \le 2\epsilon + \mathcal{O}(n\epsilon^2)$. It is more convenient to write instead $\eta = \|y\|_1/(1 + h_6)$, where $|h_6| \le 2\epsilon + \mathcal{O}(n\epsilon^2)$.

5. A final division completes the normalization,

$$\hat{x}^{(k+1)} = \mathrm{fl}\left(\frac{y}{\eta}\right) = \frac{y^T}{\eta} + h_7^T, \qquad \text{where} \quad |h_7^T| \le \epsilon\,\frac{y^T}{\eta}.$$

To express $\hat{x}^{(k+1)}$ in terms of $x^T G$ write

$$\hat{x}^{(k+1)} = \frac{y^T}{\|y\|_1} + \frac{y^T}{\|y\|_1} h_6 + h_7^T = \frac{x^T G + h_4}{\|y\|_1} + \frac{y^T}{\|y\|_1} h_6 + h_7^T = x^T G + g_{k+1}^T,$$

where

$$g_{k+1}^T = \frac{(1 - \|y\|_1) x^T G + h_4^T}{\|y\|_1} + \frac{y^T}{\|y\|_1} h_6 + h_7^T.$$

Apply $\|y\|_1 \ge 1 - \|h_4\|_1$ twice to get

$$\|g_{k+1}\|_1 \le \frac{2\|h_4\|_1}{1 - \|h_4\|_1} + |h_6| + \|h_7\|_1.$$

From $\|h_7\|_1 \le \epsilon(1 + |h_6|)$ and $|h_6| \le 2\epsilon + \mathcal{O}(n\epsilon^2)$ follows

$$\|g_{k+1}\|_1 \le \frac{2\|h_4\|_1}{1 - \|h_4\|_1} + 3\epsilon + \mathcal{O}(n\epsilon^2).$$

At last use $\|h_4\|_1 \le \epsilon(3.03 + c\alpha M)$. ∎

TABLE 5.1
Range of parameter values for experiments in section 6.

Unit roundoff	$\epsilon \approx 10^{-16}$
Damping factor	$\alpha = .85$
Dimension of H, S, G	$n \leq 4 \cdot 10^6$
Max # nonzeros in columns of H	$m \leq 2 \cdot 10^5$
Max # iterations	$k \leq 200$

Theorem 5.2 implies that the roundoff error in an iteration of the power method (FP) is bounded approximately by

$$\|g_k\|_1 \lessapprox \frac{2\alpha M\epsilon}{1-\alpha M\epsilon} \leq 4\alpha M\epsilon \qquad \text{if} \quad \alpha M\epsilon \leq 1/2.$$

Because we assume use of compensated summation, the roundoff error $\|g_k\|_1$ does not depend, to first order, on the matrix dimension n. It also does not depend on the iteration count k. The roundoff error is more or less constant, and the same for all iterations. It represents the error caused by a single matrix vector multiply, and is determined, for the most part, by the maximal number of nonzeros m in any column of H (i.e., the maximal number of inlinks into any web page) and the number of dangling nodes $\|d\|_1$, whichever is larger. The discussion relating to Figure 5.1 in section 5.2 indicates that the bounds in Theorem 5.2 are realistic, and not too pessimistic.

The only error we did not capture effectively in Theorem 5.2 consists of the higher order effects $\mathcal{O}(n\epsilon^2)$ in the compensated summation. Higher order effects can be completely avoided with *doubly compensated summation* [22, Algorithm 4.3] for applications of PageRank for matrices with dimensions not exceeding $n \leq 2^{13} = 8192$.

COROLLARY 5.3 (floating point version of error bounds). *With the assumptions in Theorem* 5.2*, the bounds in Theorem* 5.1 *hold with*

$$\|g_k\|_1 \leq \frac{2\epsilon(3.03 + c\alpha M)}{1-\epsilon(3.03+c\alpha M)} + \mathcal{O}(n\epsilon^2), \qquad k \geq 1,$$

where $M \equiv \max\{m, \|d\|_1 + 1\}$ *and* $c \equiv 1.01(1 + 3.03\epsilon)$.

Corollary 5.3 implies that the floating point error in the bounds is independent of the iteration count. The error is caused essentially by the matrix vector multiply and can be assumed to be constant. Moreover, the contribution of the floating point error to the different types of bounds is essentially the same, so that all bounds incur more or less the same floating point error.

We examine the ramifications of the above analysis when the power method (FP) is applied to the data matrices in section 6, whose parameter ranges are listed in Table 5.1. The simple bound in Theorem 4.3 and the roundoff error bound in Theorem 5.2 amount to $2\alpha^k \geq 10^{-14}$ and $\|g_k\|_1 \leq 4 \cdot 10^{-11}$. The roundoff error dominates the ranking bounds in later iterations, so that the bounds remain essentially constant from then on. Since the iterates can still change, though, one could continue the power method (FP) as long as the ranking criteria in Theorem 5.1 collect new ranking information. Note that the ranking criteria do not care whether the errors are due to finite termination or roundoff. For illustration purposes we execute 200 iterations of the power method in the experiments in section 6. A suitable termination criterion would stop the iterations once $\log(2\alpha^k) < \log(\|g_k\|_1)$.

TABLE 6.1
Properties of the data matrices (n = matrix dimension, m = maximal number of nonzeros in any column, $M = \max\{m, \text{dangling nodes} + 1\}$, and g = roundoff error).

Matrix	n	Nonzeros	m	Dangling nodes		M	g
H_S	9,914	36,854	340	2,861	29%	2,862	10^{-12}
H_L	3,148,440	39,383,235	168,685	91,462	3%	168,685	10^{-10}

The higher order effects $\mathcal{O}(n\epsilon^2)$ from the compensated summation are not likely to be of any consequence for the experiments in section 6 because $n\epsilon^2 \leq 10^{-25}$, which is negligible compared to $2\alpha^k \geq 10^{-14}$.

6. Numerical experiments. We present numerical experiments on data matrices from web crawls to compare the finite precision error bounds in section 5 and assess the performance of the ranking criterion. We describe the data matrices in section 6.1, and compare the bounds with respect to tightness in section 6.2 and with regard to ranking performance in section 6.3.

6.1. Data matrices. We present numerical experiments with two matrices that are obtained from web crawls and available on David Gleich's web page [16].

The properties of the two matrices are listed in Table 6.1. The small matrix H_S of dimension 9,914 represents a 2001 crawl [16, Webbase subgraph cs.stanford.edu], while the larger matrix H_L of dimension 3,148,440 represents a 2006 crawl [16, Wikipedia 2006-11-04]. Although the matrix H_S is small and dates from an older crawl, its larger percentage of dangling nodes is more representative of web graphs than that of H_L.

We choose the most popular values for the parameters of the Google matrix: $\alpha = .85$ for the amplification factor; and the uniform vector for personalization, dangling node, and starting vectors, $x^{(0)} = v = w = \frac{1}{n}\mathbf{1}$. The two data matrices for our experiments are

$$G_S \equiv \alpha(H_S + dw^T) + (1-\alpha)\mathbf{1}v^T, \qquad G_L \equiv \alpha(H_L + dw^T) + (1-\alpha)\mathbf{1}v^T.$$

The quantity g in Table 6.1 denotes the roundoff error from Corollary 5.3,

$$g \equiv \frac{2\epsilon(3.03 + c\alpha M)}{1 - \epsilon(3.03 + c\alpha M)}, \qquad M \equiv \max\{m, \|d\|_1 + 1\}, \qquad c \equiv 1.01(1 + 3.03\epsilon).$$

Note that the number of dangling nodes in G_S exceeds the maximal number of inlinks. Hence the dominant amplification factor M for the roundoff error of G_S is determined by the number of dangling nodes; see Theorem 5.2. This may reflect more what happens in practice when the nondangling nodes can be outnumbered by the dangling nodes, especially when applied to web graphs, since dangling nodes are part of the ever increasing web frontier. All experiments are performed in Matlab. We did not compute the norms with compensated summation because Matlab's accuracy appears to be sufficient for small problems with $n \leq 10^7$.

6.2. Tightness of the bounds. We compare bounds for the error $\|\hat{x}^{(k)} - \pi\|_1$ in iteration k. Since the roundoff error is essentially the same for all bounds and constant in each iteration, see Corollary 5.3, it suffices to compare the exact bounds. We assume that storage for 3 iterates is available, and that no overwriting takes place, so that successive iterates $\hat{x}^{(k-1)}$ and $\hat{x}^{(k)}$ require different storage locations. Among the resulting seven bounds below we included the k-step backward bound to illustrate the behavior of a bound where j is a function of k.

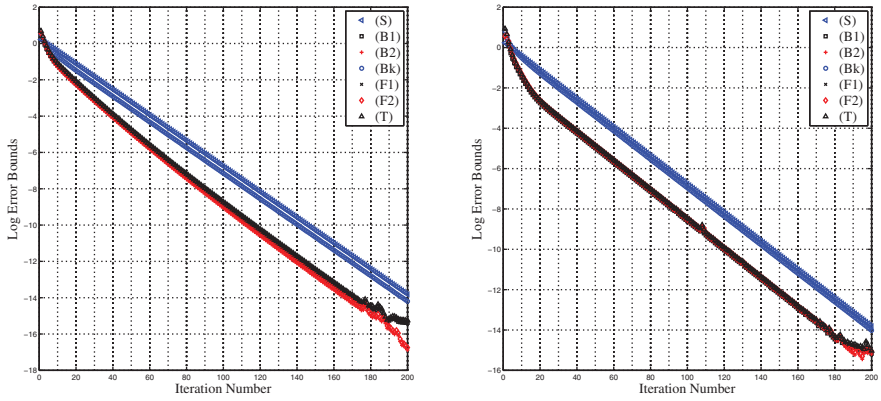


FIG. 6.1. *Error bounds for power method* (FP) *applied to the matrices G_S (left) and G_L (right).*

(S) Simple bound: $2\alpha^k$

(B1) Backward looking 1-step bound: $\frac{\alpha}{1-\alpha}\|\hat{x}^{(k-1)} - \hat{x}^{(k)}\|_1$

(B2) Backward looking 2-step bound: $\frac{\alpha^2}{1-\alpha^2}\|\hat{x}^{(k-2)} - \hat{x}^{(k)}\|_1$

(Bk) Backward looking k-step bound: $\frac{\alpha^k}{1-\alpha^k}\|x^{(0)} - \hat{x}^{(k)}\|_1$

(F1) Forward looking 1-step bound: $\frac{\|\hat{x}^{(k+1)} - \hat{x}^{(k)}\|_1}{1-\alpha}$

(F2) Forward looking 2-step bound: $\frac{\|\hat{x}^{(k+2)} - \hat{x}^{(k)}\|_1}{1-\alpha^2}$

(T) Two level 1-1-step bound: $\|\hat{x}^{(k+1)} - \hat{x}^{(k)}\|_1 + \frac{\|\hat{x}^{(k+2)} - \hat{x}^{(k+1)}\|_1}{1-\alpha}$

Figure 6.1 shows the above bounds for the matrices G_S and G_L. The bounds fall into two groups. The first group, consisting of (S) and (Bk), is less tight than the second group, which comprises the remaining bounds. There is little difference among the bounds in the second group. The straight lines in the context of the vertical logarithmic axis suggest that the geometric distances between iterates decrease at the same rate.

6.3. Ranking performance. Due to the lack of difference among the bounds in the competitive second group, we choose only (B1) for ranking, because it is the cheapest. The floating point version of the corresponding ranking criterion from Theorem 5.1 is as follows: If $\hat{x}_i^{(k)} > \hat{x}_j^{(k)} + \beta_B$, then $\pi_i > \pi_j$, where

$$\text{(B1-FP)} \qquad \beta_B \equiv \frac{\alpha}{1-\alpha}\|\hat{x}^{(k-1)} - \hat{x}^{(k)}\|_1 + g.$$

Applicability. Let Q_j be a permutation that orders the elements of $\hat{x}^{(j)}$ in decreasing order. That is, $\tilde{x}^{(j)} \equiv Q_j\hat{x}^{(j)}$ where $\tilde{x}_1^{(j)} \geq \ldots \geq \tilde{x}_n^{(j)}$. We count the number of pairs to which the criterion (B1-FP) applies. That is, we count the number of distinct pairs for which $\tilde{x}_i^{(j)} > \tilde{x}_{i+1}^{(j)} + \beta_B$ in iterations $1, \ldots, j$. Figure 6.2 shows this number for each iteration with the matrix G_S (due to memory limitations we were not able to collect this information for the large matrix G_L in every iteration). The line in the upper half represents the number of pairs of identical elements $\tilde{x}_i^{(j)} = \tilde{x}_{i+1}^{(j)}$ in each iterate. For instance, if $\tilde{x}_1^{(j)} = \tilde{x}_2^{(j)} = \tilde{x}_3^{(j)}$, then we count the two pairs $(1, 2)$ and $(2, 3)$. Since the ranking criterion cannot be applied to $\tilde{x}_1$ and $\tilde{x}_2$, the number of identical element pairs puts a natural limit on the performance of any ranking criterion that does not rely on additional criteria.

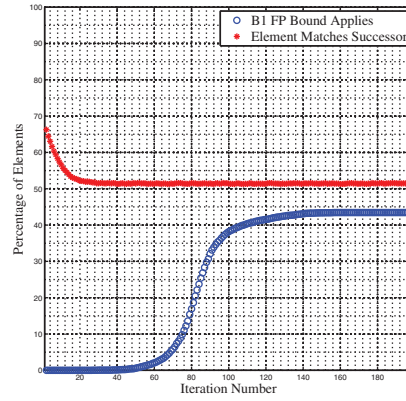


FIG. 6.2. *Applicability of ranking criterion* (B1-FP) *applied to the small matrix* G_S.

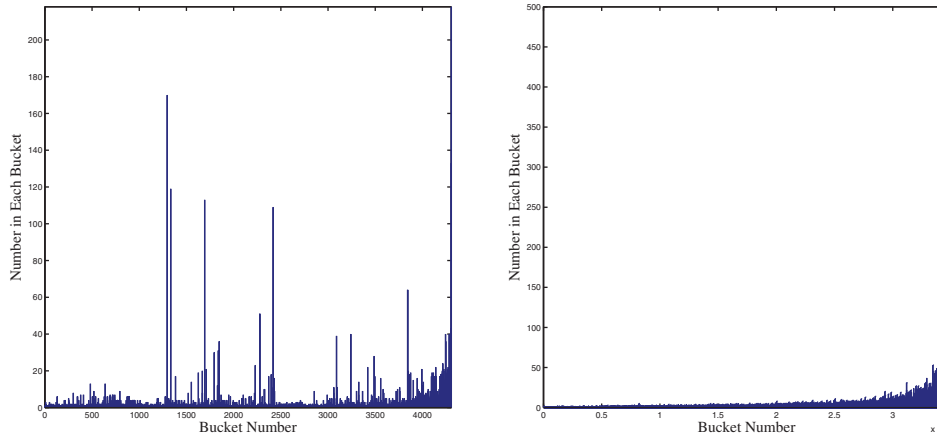


FIG. 6.3. *Buckets for ranking criterion* (B1-FP) *applied to iteration* 200 *with matrices* G_S *(left) and* G_L *(right).*

Figure 6.2 suggests that over 50% of the elements in most iterates are identical to another element. After about 100 iterations, criterion (B1-FP) applies to more than 40% of the elements. This means that only less than 10% of the elements remain unranked. The collection of new ranking information seems to level off after about 140 iterations. We can explain this as follows. The simple ranking bound $2\alpha^k$ is dominated by the roundoff error $\|g_k\|_1$ after about 170 iterations. Since the (B1) bound is tighter than the simple bound, the roundoff error dominates earlier. This is another justification for a termination criterion of the type already mentioned in section 5.3: Terminate the power method (FP) as soon as $\beta_B \approx g$.

Bucket ranking. Figure 6.3 gives an idea of how many different elements can be ranked and how big the buckets are. The histograms refer to the last iteration and depict the number of elements per bucket. To prevent distortion of the vertical axis and to assure better visibility for the top-ranked buckets, we omit the rightmost buckets (a single bucket in case of G_S and many buckets in case of G_L), which are the largest and contain the smallest elements.

Table 6.2 gives detailed information about the number and size of the buckets. The number of buckets represents *distinct* ranks that have been identified. For both

TABLE 6.2
Number and size of buckets for the matrices G_S (top) and G_L (bottom). The last two columns list the number of elements in the first and last buckets, respectively.

n	# Buckets	First bucket	Last bucket	
9,914	4,307	1	699	7%
3,148,440	34,911	1	2,996,646	95%

TABLE 6.3
Ranking information for the matrices G_S (top) and G_L (bottom). The second column lists the number of elements with exact rank, the third column the number of exactly ranked elements among the top 100 elements, and the last column lists the lowest rank that could be distinguished.

n	Exact ranking		Exact top 100	Lowest rank
9,914	3,177	32%	79	9,215
3,148,440	24,120	0.76%	100	151,794

matrices the ranking criterion (B1-FP) isolates the highest ranked PageRank score, because the first bucket contains only a single element. For the small matrix G_S, 7% of the smallest elements cannot be ranked, while for the large matrix G_L this number increases to 95%.

Table 6.3 gives detailed information about the ranking. It shows how many elements are ranked exactly, how many elements among the top 100 are exactly ranked, and the lowest rank that could be identified. Since the lowest identified rank for the matrix G_S is 9,215, the smallest bucket contains $n - 9,215 = 699$ elements, as shown in Table 6.2. The preceding information illustrates that the ranking criteria are able to identify the PageRanks of the top-ranked elements.

7. Extremely large matrices. As stated in Theorem 3.1, the main idea of our paper is the following ranking criterion: Let $x \geq 0$ with $\|x\|_1 = 1$ be any approximation to the PageRank vector π, and $\beta \geq \|x - \pi\|_1$.

(C) $$\text{If } x_i > x_j + \beta, \text{ then } \pi_i > \pi_j.$$

In the preceding sections we discussed the performance of (C) for matrices of dimension $n \leq 4 \cdot 10^6$, and for many applications of PageRank this is sufficient. However, the indexed web comprises hundreds of billions of web pages. Below are several suggestions for how to apply the criterion to matrices of extreme dimension.

7.1. Curbing roundoff error. The subsequent discussions are based on the roundoff error analysis in Theorem 5.2, which is valid for matrix dimensions $n < 10^{14}$ in IEEE double precision. Our experiments suggest that these roundoff error bounds are accurate and not at all pessimistic.

1. Computation of iterate norms. To prevent first-order dependence of the roundoff error on n, the iterates must be normalized on a regular basis, and the norms computed with compensated summation. However, even with compensated summation the higher order terms of the roundoff error can reach $n\epsilon^2 = 10^{-18}$ for $n = 10^{14}$ in IEEE double precision, which is large enough to be of concern for the ranking bound (C). Doubly compensated summation, or cascaded compensated summation [35, Algorithm 4.8], may be able to reduce higher order effects.

2. Matrix vector multiplications. Theorem 5.2 shows that with accurate computation of the iterate norms, the roundoff error is mainly due to a single matrix vector multiplication. In particular, it is determined by the maximal number of nonzeros in any column of the web matrix H (maximal number of inlinks) and the number of

dangling nodes (web pages without outlinks), whichever is larger. Since the dangling nodes are part of the increasing web frontier, they can easily outnumber the inlinks [12, section 2] and contribute substantially to the roundoff error. In section 7.2 we indicate how to reduce the influence of the dangling nodes.

3. Termination criteria. In later iterations the ranking bound β in (C) is dominated by roundoff error. Once this happens no new ranking information seems to be available. One may want to terminate the power method as soon as β is on the order of the roundoff error. The bound β is computed from geometric differences between iterates, that is, expressions of the form $\|\hat{x}^{(k+j)} - \hat{x}^{(k)}\|_1$. Catastrophic cancellation may damage the accuracy of these norms. This can be circumvented by resorting to the simple bound $2\alpha^k$ in Theorem 4.3. However, this bound is the least tight among all the bounds and requires the most iterations. A practical approach might be to just iterate until β is on the order of the roundoff error, so that accuracy in the computation of $\|\hat{x}^{(k+j)} - \hat{x}^{(k)}\|_1$ becomes less important, and collect ranking information only in the very last iteration.

7.2. Reducing matrix dimension. There are at least two advantages in reducing the matrix dimension: Faster computation and smaller roundoff error. Two easy approaches involve eliminating unreferenced pages (pages without inlinks) and dangling nodes (pages without outlinks). After such a reduction, the computation time depends only on the number of nondangling referenced pages, and the roundoff error depends only on the maximal number of inlinks to nondangling nodes.

1. Unreferenced pages. Suppose, as is likely in applications of PageRank to web graphs, that the dangling nodes outnumber the unreferenced pages, and that we have reordered the web matrix H so that the unreferenced pages are numbered last,

$$QHQ^T = \begin{pmatrix} H_1 & H_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where the diagonal blocks are square and Q is a permutation matrix. If one sets the trailing block of the dangling node vector equal to zero $w^TQ^T = \begin{pmatrix} w_1^T & w_2^T & 0 \end{pmatrix}$ so that the dangling nodes do not add new inlinks to the unreferenced pages, then the last block column of S is also zero. If we partition the personalization vector conformally, $v^TQ^T = \begin{pmatrix} v_1^T & v_2^T & v_3^T \end{pmatrix}$. Then $\pi^T = \pi^T G$ implies that the PageRank of the unreferenced pages is simply $(1-\alpha)v_3$. Furthermore, setting $v_3 = 0$ forces the PageRank of the unreferenced pages to be zero, so that they automatically receive the lowest ranking. Therefore, by keeping the dangling node vector positions associated with unreferenced pages zero, one can compute the PageRank of the remaining pages from the smaller matrix $\left(\begin{smallmatrix} H_1 & H_2 \\ 0 & 0 \end{smallmatrix}\right)$.

2. Dangling nodes. One can further reduce the matrix dimension by lumping all dangling nodes into a single node. The resulting lumped matrix $S_l \equiv \left(\begin{smallmatrix} H_1 & H_2\mathbf{1} \\ w_1^T & w_2^T\mathbf{1} \end{smallmatrix}\right)$ is stochastic and its dimension is equal to one plus the number of nondangling nodes [23]. One can rank the nondangling nodes by applying the power method (FP) and the ranking criterion (C) to $G_l \equiv \alpha S_l + (1-\alpha)\mathbf{1}v_l^T$, where $v_l^T \equiv \begin{pmatrix} v_1^T & v_2^T\mathbf{1} \end{pmatrix}$. The PageRanks of the dangling nodes can be recovered with a single matrix vector multiplication [23].

7.3. Ranking with faster converging methods. The ranking criterion (C) is not tied to any computational method. To apply it to a method other than the power method, one first needs rigorous bounds on the forward error $\|x - \pi\|_1$ that also

take into account roundoff error. This may be hard to do for methods with involved decision processes and intricate and possibly worse conditioned matrix operations.

Instead it may be easier to compute an approximation to PageRank with a fast converging method, such as a Krylov space method [17, 18], and then use this approximation as a restart for a single power method iteration to rank the nondangling nodes. Here is a more detailed description.

1. Apply a fast method to compute an approximation z to the PageRank of the lumped matrix G_l, and terminate when the residual norm is less than g_l. Here

$$g_l \equiv \frac{2\epsilon(3.03 + c\alpha m_l)}{1 - \epsilon(3.03 + c\alpha m_l)}, \qquad c \equiv 1.01(1 + 3.03\epsilon)$$

is the roundoff error in a single iteration of the power method, and m_l is the maximal number of nonzeros in any column of S_l.

2. Execute a single iteration of the power method (FP) with $\hat{x}^{(0)} := z/\|z\|_1$ as the starting vector. That is, $[y^{(1)}]^T := [\hat{x}^{(0)}]^T G_l$, $\hat{x}^{(1)} := y^{(1)}/\|y^{(1)}\|_1$, and compute $\|z\|_1$ and $\|y^{(1)}\|_1$ by a cacaded compensated summation method [35] or by doubly compensated summation [22, Algorithm 4.3].

3. Determine the ranking bound (B1-FP) from section 6.3, $\beta_l := \|\hat{x}^{(1)} - \hat{x}^{(0)}\|_1 + g_l$, where $\|\hat{x}^{(1)} - \hat{x}^{(0)}\|_1$ is computed by doubly compensated summation. Use the ranking criterion: If $\hat{x}_i^{(1)} > \hat{x}_j^{(1)} + \beta_l$, then $\pi_i > \pi_j$, and construct buckets according to the rules in section 3.

Acknowledgments. We thank Steve Kirkland for many helpful discussions, and David Gleich for sharing his data so generously and patiently. We are also grateful to Nick Higham for suggesting the use of compensated summation.

REFERENCES

[1] *http://www.google.com/technology/*. Our Search: Google Technology.

[2] R. Andersen, F. R. K. Chung, and K. J. Lang, *Local partitioning for directed graphs using PageRank*, in Algorithms and Models for the Web-Graph, Lecture Notes Comput. Sci. 4863, Springer Verlag, New York, 2007, pp. 166–178.

[3] R. Baeza-Yates, P. Boldi, and C. Castillo, *Generalizing PageRank: Damping functions for link-based ranking algorithms*, in Proc. ACM SIGIR, Seattle, WA, August 2006, ACM Press, pp. 308–315.

[4] P. Berkhin, *A survey on PageRank computing*, Internet Math., 2 (2005), pp. 73–120.

[5] M. Bianchini, M. Gori, and F. Scarselli, *Inside PageRank*, ACM Trans. Internat. Tech., 5 (2005), pp. 92–128.

[6] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas, *Link analysis ranking: Algorithms, theory, and experiments*, ACM Trans. Internat. Tech., 5 (2005), pp. 231–297.

[7] C. Brezinski and M. Redivo-Zaglia, *The PageRank vector: Properties, computation, approximation, and acceleration*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 551–575.

[8] S. Brin and L. Page, *The anatomy of a large-scale hypertextual Web search engine*, Comput. Networks ISDN Syst., 30 (1998), pp. 107–117.

[9] P. Chen, H. Xie, S. Maslov, and S. Redner, *Finding scientific gems with Google's PageRank algorithm*, J. Infornmet., 1 (2007), pp. 8–15.

[10] J. V. Davis and I. S. Dhillon, *Estimating the global PageRank of Web communities*, in KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, 2006, pp. 116–125.

[11] R. Dellavalle, L. Schilling, M. Rodriguez, H. V. de Sompel, and J. Bollen, *Refining dermatology journal impact factors using PageRank*, J. Am. Acad. Dermatol., 57 (2007), pp. 116–119.

[12] N. Eiron, K. S. McCurley, and J. A. Tomlin, *Ranking the Web frontier*, in WWW '04: Proceedings of the 13th Conference on World Wide Web, ACM Press, New York, 2004, pp. 309–318.

[13] L. Eldén, *A note on the eigenvalues of the Google matrix*, Technical report, LiTH-MAT-R-04-01, Department of Mathematics, Linköping University, 2004.
[14] A. Esuli and F. Sebastiani, *PageRanking WordNet synsets: An application to opinion mining*, in Proceedings of ACL 2007, Prague, CZ, 2007, pp. 25–27.
[15] R. Fagin, R. Kumar, and D. Sivakumar, *Comparing top* k *lists*, SIAM J. Discrete Math., 17 (2003), pp. 425–431.
[16] D. Gleich, *http://www.stanford.edu/~dgleich/data/*. Datasets, graphs and more.
[17] D. Gleich, L. Zhukov, and P. Berkhin, *Fast parallel PageRank: A linear system approach*, Technical report, Yahoo! Inc., 2004.
[18] G. H. Golub and C. Greif, *An Arnoldi-type algorithm for computing PageRank*, BIT, 46 (2006), pp. 759–771.
[19] T. L. Griffiths, M. Steyvers, and A. Firl, *Google and the mind: Predicting fluency with PageRank*, Psychological Science, 18 (2007), pp. 1069–1076.
[20] T. H. Haveliwala and S. D. Kamvar, *The second eigenvalue of the Google matrix*, Technical report 2003-20, Stanford University, 2003.
[21] M. Haviv and L. V. D. Heyden, *Perturbation bounds for the stationary probabilities of a finite Markov chain*, Adv. Appl. Prob., 16 (1984), pp. 804–818.
[22] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[23] I. C. F. Ipsen and T. M. Selee, *PageRank computation, with special attention to dangling nodes*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1281–1296.
[24] I. C. F. Ipsen and R. S. Wills, *Mathematical properties and analysis of Google's PageRank*, Bol. Soc. Esp. Mat. Apl., 34 (2006), pp. 191–196.
[25] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, *Exploiting the block structure of the Web for computing PageRank*, Technical report 2003-17, Stanford University, 2003.
[26] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, *Extrapolation methods for accelerating PageRank computations*, in The Twelfth International World Wide Web Conference, ACM Press, New York, 2003, pp. 261–270.
[27] S. M. G. Kendall, *Rank Correlation Methods*, Charles Griffin & Company Limited, 1975.
[28] A. Khalil and Y. Liu, *Experiments with PageRank computation*, Technical report 603, Computer Science Department at Indiana University, 2004.
[29] S. J. Kirkland, *Conditioning of the entries in the stationary vector of a Google-type matrix*, Linear Algebra Appl., 418 (2006), pp. 665–681.
[30] S. M. Kirsch, M. Gnasa, and A. B. Cremers, *Beyond the Web: Retrieval in social information spaces*, in ECIR, 2006, pp. 84–95.
[31] A. N. Langville and C. D. Meyer, *A survey of eigenvector methods for Web information retrieval*, SIAM Rev., 47 (2005), pp. 135–161.
[32] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ, 2006.
[33] R. Mihalcea, P. Tarau, and E. Figa, *PageRank on semantic networks, with application to word sense disambiguation*, in COLING '04: Proceedings of the 20th International Conference on Computational Linguistics, Morristown, NJ, 2004, Association for Computational Linguistics.
[34] J. L. Morrison, R. Breitling, D. J. Higham, and D. R. Gilbert, *GeneRank: Using search engine technology for the analysis of microarray experiments*, BMC Bioinformatics, 6 (2005), p. 233.
[35] T. Ogita, S. M. Rump, and S. Oishi, *Accurate sum and dot product*, SIAM J. Sci. Comput., 26 (2005), pp. 1955–1988.
[36] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank citation ranking: Bringing order to the Web*, Technical report, Stanford University, 1998.
[37] S. P. Ponzetto and M. Strube, *Deriving a large scale taxonomy from Wikipedia*, in AAAI '07, 2007, pp. 1440–1445.
[38] F. Qiu and J. Cho, *Automatic identification of user interest for personalized search*, in WWW '06: Proc. 15th International Conference on World Wide Web, ACM Press, New York, 2006, pp. 727–736.
[39] T. Sarlós, A. A. Benczúr, K. Csalogány, D. Fogaras, and B. Rácz, *To randomize or not to randomize: Space optimal summaries for hyperlink analysis*, in WWW '06: Proc. 15th International Conference on World Wide Web, ACM Press, New York, 2006, pp. 297–306.
[40] S. Serra-Capizzano, *Jordan canonical form of the Google matrix: A potential contribution to the PageRank computation*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 305–312.
[41] T. Simonite, *Google tool could search out hospital superbugs*, 30 January 2008.

[42] G. W. Stewart, *On the powers of a matrix with perturbations*, Numer. Math., 96 (2003), pp. 363–376.
[43] R. S. Varga, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1962.
[44] P. von Hilgers and A. N. Langville, *The five greatest applications of Markov Chains*, in Proceedings of the Markov Anniversary Meeting, Boston Press, Boston, MA, 2006.
[45] J. Wang, J. Liu, and C. Wang, *Keyword extraction based on PageRank*, in PAKDD, 2007, pp. 857–864.
[46] Y. Wang and D. J. DeWitt, *Computing PageRank in a distributed Internet search system*, in Proc. 30th VLDB Conference, 2004.
[47] R. S. Wills, *When rank trumps precision: Using the power method to compute Google's PageRank*, http://www.lib.ncsu.edu/theses/available/etd-06122007-173712/, 2007.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1697–1708

VALIDATED SOLUTIONS OF SADDLE POINT LINEAR SYSTEMS*

TAKUMA KIMURA† AND XIAOJUN CHEN‡

Abstract. We propose a fast verification method for saddle point linear systems where the (1,1) block is singular. The proposed verification method is based on an algebraic analysis of a block diagonal preconditioner and rounding mode controlled computations. Numerical comparison of several verification methods with various block diagonal preconditioners is given.

Key words. saddle point matrix, numerical verification, block preconditioning

AMS subject classifications. 65F05, 65G05, 65G20

DOI. 10.1137/070706441

1. Introduction.

We consider the system of saddle point linear systems

$$\mathcal{H}u = b \equiv \begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}, \tag{1.1}$$

where A is an $n \times n$ symmetric positive semidefinite matrix and B is an $n \times m$ matrix, with $m \leq n$. We denote $l = n + m$. We assume that the coefficient matrix $\mathcal{H}$ is nonsingular, which implies that B has full-column rank.

In recent years, saddle point problems have received considerable attention. A large amount of work has been devoted to developing efficient algorithms for solving saddle point problems. In a recent comprehensive survey [1], Benzi, Golub, and Liesen discussed a large selection of numerical methods for saddle point problems. We are aware that it is very important to verify the accuracy of approximate solutions obtained by the numerical methods. However, there is little discussion on validated solutions of saddle point problems by taking all possible effects of rounding errors into account.

Standard validation methods for the solution of a system of linear equations use an approximation of the inverse of the coefficient matrix. These methods are not efficient for the saddle point problem (1.1) when the dimension l is large or the condition number of $\mathcal{H}$ is large, due to the indefiniteness of $\mathcal{H}$ and the singularity of A.

In [2], a numerical validation method is proposed for verifying the accuracy of approximation solutions of the saddle point problem (1.1) without using an approximation of the inverse $\mathcal{H}^{-1}$, under the assumption that A is symmetric positive definite. The method uses the special structure of the saddle point problem to represent the variable x by the inverse of A and the variable y. For the case that A is singular and the size of the problem is large, it is a significant challenge to compute a rigorous upper bound for the norm $\|\mathcal{H}^{-1}\|$ without using an approximation of the inverse $\mathcal{H}^{-1}$. In this paper, we present a fast method to compute rigorous error bounds for the

*Received by the editors October 26, 2007; accepted for publication (in revised form) by A. J. Wathen September 18, 2008; published electronically January 23, 2009. The first version of this paper was completed while the second author was working at Hirosaki University. This work was supported partly by the Scientific Research Grant-in-Aid from the Japan Society for Promotion of Science.
http://www.siam.org/journals/simax/30-4/70644.html

†Department of Mathematical Sciences, Hirosaki University, Hirosaki, 036-8561, Japan (h07gs801@stu.hirosaki-u.ac.jp).

‡Department of Applied Mathematics, The Hong Kong Polytechnic University, Kowloon, Hong Kong, People's Republic of China (maxjchen@polyu.edu.hk). This author's work was supported partly by the Research Grants Council of Hong Kong.

saddle point problem (1.1) where A is symmetric positive semidefinite. In particular, we present a fast method to compute a constant α such that

$$\|u^* - u\|_2 \le \alpha \|b - \mathcal{H}u\|_2 \quad \text{for } u \in R^l, \tag{1.2}$$

where u^* is the exact solution of (1.1). This method is based on an algebraic analysis of a block diagonal preconditioner for saddle point systems studied in a recent paper [6] by Golub, Greif, and Varah. Instead of approximating the inverse of the $l \times l$ indefinite matrix $\mathcal{H}$, we use approximations of inverses of two symmetric positive definite matrices in $R^{n\times n}$ and $R^{m\times m}$ to define the constant α in the error bound (1.2). Moreover, we present fast methods to estimate upper bounds for the norms of inverses of the two symmetric positive definite matrices based on fast validated matrix computation by Oishi and Rump [8].

In section 2, we define the error constant α. In section 3, we discuss how to compute an upper bound of α efficiently and accurately by taking all possible effects of rounding errors into account. In section 4, we compare our verification method with the Krawczyk method [13], the LU decomposition method [8], the verifylss function of INTLAB, and a block component verification method proposed in [2] using examples from CUTEr [6], optimal surface fitting [3, 10], mixed finite element discretization of the Stokes equations [1], and image restoration [5, 15].

2. A new error bound. Let W be an $m \times m$ symmetric positive semidefinite matrix such that

$$M(W) = A + BWB^T$$

is a symmetric positive definite matrix. Note that BWB^T is singular for any symmetric positive definite matrix W when $m < n$. However, if W is symmetric positive definite, we can show that $M(W)$ is symmetric positive definite under the conditions that A is positive semidefinite and $\mathcal{H}$ is nonsingular. To see it, let $\bar{x} \neq 0$ be a solution of $M(W)\bar{x} = 0$. Then we have

$$\bar{x}^T A\bar{x} + \bar{x}^T BWB^T\bar{x} = 0.$$

Since A and BWB^T are positive semidefinite, we obtain

$$\bar{x}^T A\bar{x} = 0 \quad \text{and} \quad \bar{x}^T BWB^T\bar{x} = 0,$$

which implies

$$A\bar{x} = 0 \quad \text{and} \quad B^T\bar{x} = 0.$$

Therefore, we find that $\mathcal{H}z = 0$ with $z = (\bar{x}, 0)$. This contradicts that $\mathcal{H}$ is nonsingular.

Recently, Golub, Greif, and Varah [6] performed an algebraic study of the block diagonal positive definite preconditioner

$$\mathcal{M}(W) = \begin{pmatrix} M(W) & 0 \\ 0 & B^T M(W)^{-1} B \end{pmatrix}. \tag{2.1}$$

They showed that $\mathcal{M}(W)$ has the attractive property that the eigenvalues of the associated preconditioned matrix $\mathcal{M}(W)^{-1}\mathcal{H}$ are bounded in a small range.

LEMMA 2.1 (see [6]). *The eigenvalues of the preconditioned matrix $\mathcal{M}(W)^{-1}\mathcal{H}$ are bounded within the two intervals*

$$\left[-1, \frac{1-\sqrt{5}}{2}\right] \cup \left[1, \frac{1+\sqrt{5}}{2}\right].$$

Lemma 2.1 makes it possible for us to present rigorous error bounds for the saddle point problem (1.1).

THEOREM 2.1. *Let u^* be the exact solution of* (1.1). *For any $u \in R^l$, we have*

$$\|u^* - u\|_2 \le \frac{2}{\sqrt{5}-1} \max\left(\left\|M(W)^{-1}\right\|_2, \|M(W)\|_2 \left\|\left(B^T B\right)^{-1}\right\|_2 \right) \|b - \mathcal{H}u\|_2. \tag{2.2}$$

Proof. Obviously, we have

$$\|u^* - u\|_2 \le \left\|\mathcal{H}^{-1}\right\|_2 \|b - \mathcal{H}u\|_2.$$

Now, we use Lemma 2.1 to give an upper bound of $\|\mathcal{H}^{-1}\|_2$. Let $\mathcal{L}$ be a nonsingular matrix such that $\mathcal{L}\mathcal{L}^T = \mathcal{M}(W)$, and let

$$\mathcal{S} = \mathcal{L}^{-1}\mathcal{H}\mathcal{L}^{-T}. \tag{2.3}$$

Then the inverse $\mathcal{H}^{-1}$ can be given as

$$\mathcal{H}^{-1} = \mathcal{L}^{-T}\mathcal{S}^{-1}\mathcal{L}^{-1}.$$

Since $\mathcal{H}$ and $\mathcal{S}$ are symmetric, we have

$$\begin{aligned}
\left\|\mathcal{H}^{-1}\right\|_2 &= \max_{\substack{v \in R^l \\ v \neq 0}} \left| \frac{v^T \mathcal{L}^{-T}\mathcal{S}^{-1}\mathcal{L}^{-1} v}{v^T v} \right| \\
&= \max_{\substack{v \in R^l \\ v \neq 0}} \left| \frac{v^T \mathcal{L}^{-T}\mathcal{S}^{-1}\mathcal{L}^{-1} v}{v^T \mathcal{L}^{-T}\mathcal{L}^{-1} v} \frac{v^T \mathcal{L}^{-T}\mathcal{L}^{-1} v}{v^T v} \right| \\
&\le \max_{\substack{w \in R^l \\ w \neq 0}} \left| \frac{w^T \mathcal{S}^{-1} w}{w^T w} \right| \max_{\substack{v \in R^l \\ v \neq 0}} \left| \frac{v^T \mathcal{M}(W)^{-1} v}{v^T v} \right| \\
&= \left\|\mathcal{S}^{-1}\right\|_2 \left\|\mathcal{M}(W)^{-1}\right\|_2. \qquad (2.4)
\end{aligned}$$

From (2.3) and $\mathcal{L}\mathcal{L}^T = \mathcal{M}(W)$, we have

$$\mathcal{L}^T \mathcal{M}(W)^{-1} \mathcal{H}\mathcal{L}^{-T} = \mathcal{S}.$$

Hence $\mathcal{S}$ and $\mathcal{M}(W)^{-1}\mathcal{H}$ have the same eigenvalues.

By Lemma 2.1, the eigenvalues of $\mathcal{S}$ are bounded within the two intervals

$$\left[-1, \frac{1-\sqrt{5}}{2}\right] \cup \left[1, \frac{1+\sqrt{5}}{2}\right].$$

Hence all eigenvalues of $\mathcal{S}^{-1}$ satisfy

$$|\lambda_i| \le \frac{2}{\sqrt{5}-1}, \quad i = 1, 2, \ldots, l.$$

From (2.4), we obtain

$$\left\|\mathcal{H}^{-1}\right\|_2 \le \frac{2}{\sqrt{5}-1} \left\|\mathcal{M}(W)^{-1}\right\|_2.$$

Moreover, from (2.1), we have

$$\begin{aligned}\left\|\mathcal{M}(W)^{-1}\right\|_2 &\le \max\left(\left\|M(W)^{-1}\right\|_2, \left\|\left(B^T M(W)^{-1} B\right)^{-1}\right\|_2\right)\\ &\le \max\left(\left\|M(W)^{-1}\right\|_2, \|M(W)\|_2 \left\|\left(B^T B\right)^{-1}\right\|_2\right),\end{aligned}$$

where the last inequality uses

$$\begin{aligned}\lambda_{\min}\left(B^T M(W)^{-1} B\right) &= \min_{\substack{y\in R^m\\ y\neq 0}} \frac{\left(M(W)^{-1}By, By\right)}{(By, By)} \frac{\left(B^T By, y\right)}{(y, y)}\\ &\ge \lambda_{\min}\left(M(W)^{-1}\right)\lambda_{\min}(B^T B). \qquad \square\end{aligned}$$

Theorem 2.1 shows that an upper bound of the inverse $\|\mathcal{H}^{-1}\|_2$ can be obtained by computing upper bounds for the norm of inverses of two symmetric positive definite matrices with sizes of $n \times n$ and $m \times m$. When n and/or m are large, the number of its flops is much less than the methods working on the $(n+m) \times (n+m)$ matrix $\mathcal{H}$. For example, the LU decomposition method for (1.1) requires $O((n+m)^3)$ flops, but estimating $\|M(W)^{-1}\|_\infty$ and $\|(B^T B)^{-1}\|_\infty$ requires only $O(n^3) + O(m^3)$ flops, which will save $O(n^2 m + nm^2)$ flops computational cost. Moreover, since the two matrices are symmetric, we can replace $\|\cdot\|_2$ by $\|\cdot\|_\infty$ for the matrix norm in (2.2) and have

$$\|u^* - u\|_2 \le \frac{2}{\sqrt{5}-1} \max\left(\left\|M(W)^{-1}\right\|_\infty, \|M(W)\|_\infty \left\|\left(B^T B\right)^{-1}\right\|_\infty\right) \|b - \mathcal{H}u\|_2. \tag{2.5}$$

In general, (2.5) is easier to implement than (2.2).

3. Verification methods. When we apply Theorem 2.1 and other verification methods to verify the accuracy of an approximate solution of (1.1) on a computer, it is necessary to consider rounding error. The IEEE 754 arithmetic standard [4] defines the rounding modes for double precision floating point numbers. Since Intel's CPU follows this standard, the rounding modes can be used on most personal computers (PCs) and workstations. We use rounding downwards, rounding upwards, and rounding nearest to compute rigorous error bounds for (2.5). We also apply these rounding modes to the following three verification methods.

Krawczyk method [11, 13].

$$\begin{aligned}K(U) &:= u - \mathcal{R}(\mathcal{H}u - b) + (I - \mathcal{R}\mathcal{H})(U - u),\\ K(U) \subset \mathrm{int}(U) &\Rightarrow u^* \in K(U) \Rightarrow \|u^* - u\|_\infty \le \|\mathrm{radius}(U)\|_\infty,\end{aligned}$$

where $\mathcal{R}$ is an approximate inverse of $\mathcal{H}$ and U is an interval vector whose center is u.

LU decomposition method [8]. Let $\mathcal{L}\mathcal{U}$ be an approximate LU factorization of $\mathcal{H}$, that is, $\mathcal{H} \approx \mathcal{L}\mathcal{U}$.

$$\|u^* - u\|_\infty \le \frac{\left\|\mathcal{U}^{-1}\mathcal{L}^{-1}(b - \mathcal{H}u)\right\|_\infty}{1 - \left\|\mathcal{U}^{-1}\mathcal{L}^{-1}\mathcal{H} - \mathcal{I}\right\|_\infty}.$$

Block component verification method [2]. Let $u^* = (x^*, y^*)$, $r_1 = Ax + By - c$, and $r_2 = B^T x - d$.

$$\begin{aligned}\|u^* - u\|_\infty &\le \max(\|x^* - x\|_\infty, \|y^* - y\|_\infty),\\ \|x^* - x\|_\infty &\le \|A^{-1}\|_\infty (\|r_1\|_\infty + \|B\|_\infty \|y^* - y\|_\infty),\\ \|y^* - y\|_\infty &\le \|A\|_\infty \left\|\left(B^T B\right)^{-1}\right\|_\infty \left(\|r_2\|_2 + \sqrt{\left\|BB^T\right\|_\infty \left\|A^{-1}\right\|_\infty} \|r_1\|_2\right).\end{aligned}$$

verifylss of INTLAB [14].

$$U = \mathsf{verifylss}(\mathcal{H}, b),$$
$$\|u - u^*\|_\infty \le 2\|\mathrm{radius}(U)\|_\infty \quad \text{for} \quad u \in U.$$

Note that when A is singular, the block component verification method cannot be applied to (1.1) directly but to the equivalent system

$$\begin{pmatrix} A + BWB^T & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c + BWd \\ d \end{pmatrix}. \tag{3.1}$$

Obviously, the error bounds depend on the choice of W. We tested the error bounds with various W. In this paper, we consider two types of choices

$$W_1(\gamma) = \gamma / \left\|B^T B\right\|_2 I$$

and

$$W_2(\gamma) = \gamma \left(B^T B\right)^{-1}$$

if A is singular. We set $W = 0$ if A is nonsingular.

4. Numerical experiment. The numerical testing was carried out on an IBM PC (3.0 GHz Pentium 4 processor, 1GB of memory) with the use of MATLAB 7.0 and INTLAB (Version 5.4) [12, 14]. We use the function setround in INTLAB [14] to compute the error bound. The function setround allows the rounding mode of the processor to be changed between round nearest (setround(0)), round down (setround(-1)), round up (setround(1)), and round towards zero (setround(2)). To compute $\|M(W)^{-1}\|_\infty$, we first use setround(0) to compute an approximate inverse R of $M(W)$. Next we use

```
S=intval(R),
beta=abss(norm(S,inf)/(1-norm(S*M(W)-I)))
```

to get an upper bound β for $\|M(W)^{-1}\|_\infty$ as

$$\left\|M(W)^{-1}\right\|_\infty \le \frac{\|R\|_\infty}{1 - \|RM(W) - I\|_\infty} \le \beta.$$

Similarly, we apply the function setround to (2.5) to get an upper bound Γ by taking all possible effects of rounding errors into account such that

$$\|u^* - u\|_2 \le \Gamma. \tag{4.1}$$

We compare the error bound (2.5) with the Krawczyk method, the LU decomposition method, the block component verification method, and the function verifylss of INTLAB using examples from CUTEr [6], optimal surface fitting [3, 10], mixed finite element discretization of the Stokes equations [1], and image restoration [5, 15].

Example 4.1 (CUTEr matrices). We used two test problems genhs28 and gouldqp3 from the CUTEr collection [7], which were used in [6].

The genhs28 is an $(n+m)\times(n+m)$ saddle point matrix, where A is an $n \times n$ tridiagonal matrix with 2, 4, 2 along its superdiagonal, main diagonal, and subdiagonal, respectively, except $A_{1,1} = A_{n,n} = 2$. The rank of A is $n-1$. B is $n \times m$ with values 1, 2, 3 along its main diagonal, first subdiagonal, and second subdiagonal, respectively.

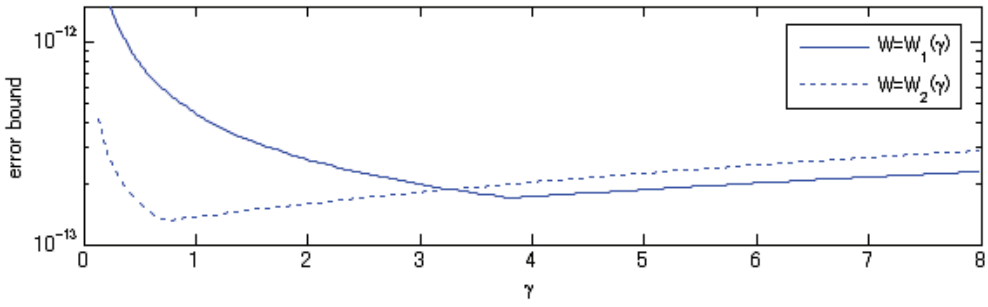


FIG. 1. *Error bounds* (2.5) *for genhs*28 *with different* W. $(n, m) = (500, 498)$, $\|A\|_2 \approx 8.0$, $\|B^T B\|_2 \approx 36.0$, *and* $\|(B^T B)^{-1}\|_2 \approx 0.38$.

TABLE 1
The error bounds for Example 4.1, *genhs*28. *rank*$(A) = n - 1$.

(n, m)		(10,8)	(500,498)	(1500,1498)	(3000,2998)
cond$(\mathcal{H})$		40.232	40.461	40.461	40.460
$\|u - u^*\|_\infty$		2.22e−16	2.22e−16	2.22e−16	9.99e−16
$\|b - \mathcal{H}u\|_2$		2.46e−14	2.02e−13	3.50e−13	1.03e−12
(2.5)	$W_1(1)$	5.67e−13 [3.6e−15]	4.58e−12 [1.859]	7.95e−12 [48.093]	2.33e−11 [477.059]
	$W_2(1)$	1.46e−13 [7.1e−15]	1.41e−12 [1.983]	2.45e−12 [48.906]	7.21e−12 [1102.207]
Block component	$W_1(1)$	6.11e−10 [3.6e−15]	5.74e−09 [1.953]	9.95e−09 [89.373]	2.82e−08 [480.387]
	$W_2(1)$	3.96e−11 [1.1e−14]	3.82e−10 [2.078]	6.63e−10 [90.185]	1.88e−09 [1105.536]
LU		5.72e−16 [3.6e−15]	5.67e−16 [14.103]	5.67e−16 [385.535]	Fail[1]
Krawczyk		2.32e−15 [0.031]	2.32e−15 [6.625]	3.55e−15 [166.894]	Fail
verifylss		3.33e−16 [0.016]	3.33e−16 [4.795]	3.33e−16 [109.827]	Fail

The gouldqp3 is an $(n + m) \times (n + m)$ saddle point matrix, where A is an $n \times n$ matrix with rank$(A) = n - 2$.

We set the exact solution u^* and right-hand side vector b as

$$u^* = (1, \ldots, 1)^T, \qquad b = \mathcal{H}u^*.$$

Figure 1 shows the error bounds for genhs28 with $(n, m) = (500, 498)$ as γ changes from $\min(\|A\|_2, 1/\|A\|_2)$ to $\max(\|A\|_2, 1/\|A\|_2)$. In Tables 1–2, we report numerical results with $W_1(1)$ and $W_2(1)$.

Example 4.2 (surface fitting problem). Let $\Omega \subset R^2$ be a convex bounded domain, $p_i = (p_i^1, p_i^2) \in \Omega$ be the measurement points, and q_i be the corresponding real values $(i = 1 \ldots k)$. We consider the following surface fitting problem [3, 10]

$$\min \quad \sum_{i=1}^{k} (f(p_i) - q_i)^2 + \mu |f|^2_{H^2(\Omega)} \tag{4.2}$$

over all functions f in the Sobolev space $H^2(\Omega)$. Here μ is a fixed parameter.

[1] In Tables 1–5, "Fail" means out of the memory; [ ] shows CPU time (sec.).

TABLE 2
The error bounds for Example 4.1, *gouldqp3. rank*$(A) = n - 2$.

(n, m)		(699,349)	(1999,999)	(2999,1499)
cond$(\mathcal{H})$		139.018	139.018	139.018
$\|u - u^*\|_\infty$		2.22e−16	2.22e−16	9.99e−16
$\|b - \mathcal{H}u\|_2$		8.29e−14	1.40e−13	3.80e−13
(2.5)	$W_1(1)$	5.63e−12 [2.406]	9.54e−12 [55.341]	2.58e−12 [194.385]
	$W_2(1)$	3.62e−12 [2.500]	6.13e−12 [81.605]	1.66e−12 [313.902]
Block component	$W_1(1)$	4.31e−09 [2.468]	7.30e−09 [55.951]	1.97e−08 [195.433]
	$W_2(1)$	1.94e−09 [2.546]	3.30e−09 [82.278]	8.90e−09 [314.996]
LU		4.75e−16 [19.420]	4.75e−16 [498.377]	Fail
Krawczyk		3.37e−15 [7.760]	3.37e−15 [179.915]	Fail
verifylss		3.33e−16 [5.968]	3.33e−16 [118.114]	Fail

We apply a finite element approximation with uniform triangular meshes to (4.2) and obtain a convex optimization problem in R^{4m+1} [3]:

$$\begin{aligned} &\min \|Nx_1 + \mu e_k - q\|_2^2 + \mu \left(x_2^T G x_2 + x_3^T G x_3\right) \\ &\text{subject to (s.t.) } Gx_1 = B_1 x_2 + B_2 x_3, \\ &\qquad\qquad e_k^T N x_1 = 0, \end{aligned} \tag{4.3}$$

where $N \in R^{k\times m}$ and $B_1, B_2, G \in R^{m\times m}$. Here G is a symmetric positive semidefinite matrix. The problem (4.3) is equivalent to the following saddle point system:

$$\left(\begin{array}{ccc|cc} N^TN & & & G & N^Te_k \\ & \mu G & & -B_1^T & \\ & & \mu G & -B_2^T & \\ \hline G & -B_1 & -B_2 & & \\ e_k^TN & & & & \end{array}\right) \left(\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ \hline y_1 \\ y_2 \end{array}\right) = \left(\begin{array}{c} 2N^T(q - \mu e_k) \\ 0 \\ 0 \\ \hline 0 \\ 0 \end{array}\right), \tag{4.4}$$

where y_1 and y_2 are the Lagrange multipliers. In many applications, $k < m$, which results in the matrix N^TN being singular; i.e., the $(1,1)$ block of the saddle point matrix in (4.4) is singular.

In this example, we use real data of the 2006 average temperature in the Aomori region from the Japan Meteorological Agency. We choose $k = 47$, $n = 3m + 1$, and rank$(A) = 2m+47$ and set $\mu =$1.0e−5 in (4.4). Numerical results of the error estimate with various m, n are given in Table 3.

Example 4.3 (the Stokes equation). We consider the saddle point system arising from the mixed finite element discretization of the stationary Stokes equation:

$$\begin{cases} -\nu\Delta u + \nabla p = \varphi & \text{in} \quad \Omega, \\ -\operatorname{div} u = 0 & \text{in} \quad \Omega, \\ u = 0 & \text{on} \quad \partial\Omega, \end{cases} \tag{4.5}$$

where $\Omega = (0,1) \times (0,1)$, $\partial\Omega$ is the boundary of Ω, $\nu > 0$ is the kinematic viscosity coefficient, $\varphi = (\varphi_1, \varphi_2)$ is a given force field, $u : \Omega \to R^2$ is a velocity field, and

TABLE 3
The error bounds for Example 4.2, surface fitting problem. rank(A) = 2m + 47.

(n,m)		(766,255)	(1450,483)	(2350,783)	(3070,1023)
cond($\mathcal{H}$)		7.22e+05	1.31e+06	2.11e+06	2.56e+06
$\|b-\mathcal{H}u\|_2$		1.41e−14	1.61e−14	2.43e−14	2.54e−14
(2.5)	$W_1(1)$	4.46e−09 [2.749]	1.20e−08 [16.154]	3.38e−08 [66.213]	4.94e−08 [145.311]
	$W_2(1)$	4.85e−09 [2.812]	3.38e−09 [16.591]	9.25e−08 [68.119]	1.27e−08 [150.388]
Block component	$W_1(1)$	4.14e+01 [2.796]	5.36e+02 [16.310]	4.74e+03 [66.415]	1.04e+04 [145.951]
	$W_2(1)$	4.51e+01 [2.859]	1.02e+02 [16.763]	6.98e+02 [68.492]	1.27e+04 [151.122]
LU		8.46e−15 [19.422]	6.98e−15 [122.702]	3.51e−14 [514.185]	Fail
Krawczyk		1.43e−13 [7.243]	2.12e−13 [27.809]	4.76e−13 [198.123]	Fail
verifylss		2.93e−14 [5.047]	4.17e−14 [29.984]	7.84e−14 [124.562]	Fail

$p : \Omega \to R$ is a pressure field. We apply a mixed finite element approximation with uniform triangular meshes and obtain a saddle point linear system (1.1) where the velocity is approximated by the standard piecewise quadratic basis functions and the pressure is approximated by piecewise linear basis functions.

In this example, A is nonsingular and $A^{-1} \geq 0$. We use Theorem 2 in [9] to get a upper bound of $\|A^{-1}\|_\infty$, i.e.,

$$\left\|A^{-1}\right\|_\infty \leq \frac{\|\tilde{w}\|_\infty}{1-\|s\|_\infty},$$

where $\tilde{w}$ is an approximate solution of $Aw = e_n$, $s = A\tilde{w} - e_n$, and $e_n = (1,1,\ldots,1)^T \in R^n$.

In this example, $\|(B^TB)^{-1}\|_2 = O(h^{-2})$, where h is the mesh size. To avoid using $\|(B^TB)^{-1}\|_2$ for small h, we consider a preconditioned system. Let L be an $m \times m$ nonsingular matrix such that $L^TL \approx B^TB$. Let

$$\mathcal{P} = \begin{pmatrix} I & \\ & L^{-1} \end{pmatrix}, \quad \tilde{\mathcal{H}} = \mathcal{P}^T\mathcal{H}\mathcal{P} = \begin{pmatrix} A & \tilde{B} \\ \tilde{B}^T & 0 \end{pmatrix}, \quad \tilde{b} = \mathcal{P}^Tb,$$
$$\tilde{B} = BL^{-1}, \quad \tilde{M}(W) = A + \tilde{B}W\tilde{B}^T.$$

Applying Theorem 2.1 to the preconditioned system

$$\tilde{\mathcal{H}}\mathcal{P}^{-1}u = \tilde{b},$$

we obtain

$$\begin{aligned}\|u-u^*\|_\infty &\leq \|\mathcal{P}\|_\infty \left\|\mathcal{P}^{-1}(u^*-u)\right\|_\infty \leq \|\mathcal{P}\|_\infty \left\|\mathcal{P}^{-1}(u^*-u)\right\|_2 \\ &\leq \frac{2\|\mathcal{P}\|_\infty}{\sqrt{5}-1} \max\left(\left\|\tilde{M}(W)^{-1}\right\|_2, \|\tilde{M}(W)\|_2 \left\|\left(\tilde{B}^T\tilde{B}\right)^{-1}\right\|_2\right) \|\mathcal{P}^T(b-\mathcal{H}u)\|_2 \\ &\leq \frac{2\max\left(1, \left\|L^{-1}\right\|_\infty\right)}{\sqrt{5}-1} \max\left(\left\|\tilde{M}(W)^{-1}\right\|_\infty, \|\tilde{M}(W)\|_\infty \left\|L\left(B^TB\right)^{-1}L^T\right\|_\infty\right) \\ &\quad \times \left\|\mathcal{P}^T(b-\mathcal{H}u)\right\|_2. \qquad (4.6)\end{aligned}$$

TABLE 4
The preconditioned error bounds for Example 4.3*, Stokes equation.* $rank(A) = n$.

(n, m)	(882,144)	(2738,400)	(9522,1296)	(20402,2704)
$\mathrm{cond}(\mathcal{H})$	4.11e+05	1.21e+06	4.09e+06	8.66e+06
$\|b - \mathcal{H}u\|_2$	1.58e−15	2.56e−15	5.04e−15	7.32e−15
(4.7)	1.10e−11 [0.078]	9.24e−11 [1.001]	1.14e−09 [19.312]	5.11e−09 [149.383]
Preconditioned block component	7.24e−11 [0.078]	6.06e−10 [1.002]	7.46e−09 [19.295]	4.59e−08 [149.352]
LU	7.04e−16 [15.780]	1.53e−15 [430.490]	Fail	Fail
Krawczyk	2.00e−14 [2.516]	6.00e−14 [50.391]	Fail	Fail
verifylss	1.11e−16 [5.202]	2.22e−16 [124.224]	Fail	Fail

We call (4.6) a preconditioned error bound. From $\|L^{-1}\|_\infty \approx \sqrt{\|(B^TB)^{-1}\|_\infty} = O(h^{-1})$, the preconditioned error bound is expected to be sharper than (2.5) for the Stokes equation. Similarly, we can get a preconditioned block component verification method as Method 2 in [2].

Numerical results of the preconditioned error bounds for Example 4.3 with $\nu = 1$ are given in Table 4.

Example 4.4 (image restoration). Suppose the discretized scenes have $p = p_1 \times p_2$ pixels. Let $f \in R^p, g \in R^q$ be the underlying image and the observed image, respectively. Let $H \in R^{q\times p}$ be the corresponding blurring matrix of block Toeplitz with Toeplitz blocks. Restoration of f is an ill-conditioned problem. We consider the linear least squares problem with Tikhonov's regularization [5, 15]

$$\min_f \|Hf - g\|_2^2 + \alpha\|Df\|_2^2, \tag{4.7}$$

where α is a regularization parameter and $D \in R^{(2p-p_1-p_2)\times p}$ is a regularization matrix of a first order finite difference operator

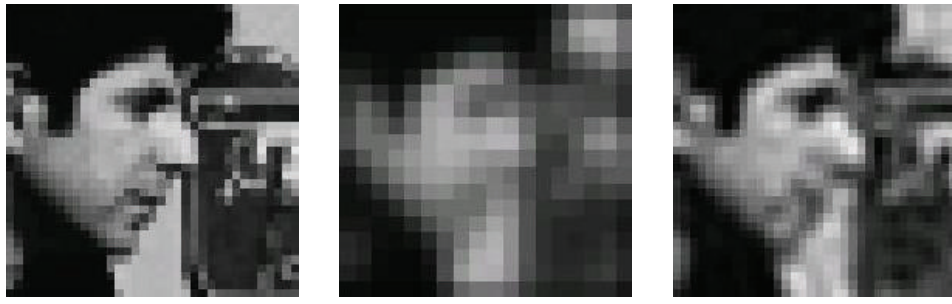
$$D = \begin{pmatrix} \Delta_{(p_1-1)\times p_1} \otimes I_{p_2\times p_2} \\ I_{p_1\times p_1} \otimes \Delta_{(p_2-1)\times p_2} \end{pmatrix} \quad \text{with} \quad \Delta = \begin{pmatrix} 1 & -1 & & \\ & 1 & -1 & \\ & & \ddots & \ddots \\ & & & 1 \quad -1 \end{pmatrix}.$$

Problem (4.7) can be rewritten as a quadratic programming

$$\begin{aligned} \min \quad & \frac{1}{2}x^TAx + c^Tx \\ \text{s.t.} \quad & B^Tx = 0, \end{aligned} \tag{4.8}$$

where

$$x = \begin{pmatrix} f \\ v \end{pmatrix}, \quad A = \begin{pmatrix} 2H^TH & 0 \\ 0 & \alpha I \end{pmatrix} \in R^{(3p-p_1-p_2)\times(3p-p_1-p_2)},$$
$$B^T = \begin{pmatrix} D & -I \end{pmatrix} \in R^{(2p-p_1-p_2)\times(3p-p_1-p_2)}, \quad c = \begin{pmatrix} -2H^Tg \\ 0 \end{pmatrix}.$$



(a) (b) (c)

FIG. 2. (a) *original image,* (b) *observed image,* (c) *restored image, PSNR* = 29.82 *db.*

TABLE 5
The error bounds for Example 4.4*, image restoration.*

(n,m)		(1160,760)	(1825,1200)	(2465,1624)	(3400,2244)
cond($\mathcal{H}$)		3.15e+08	3.40e+08	3.67e+08	4.57e+08
$\|b-\mathcal{H}u\|_2$		2.13e−14	2.80e−14	3.39e−14	4.04e−11
(2.5)	$W_1(1)$	3.01e−06 [13.564]	4.33e−06 [50.798]	5.70e−06 [124.061]	6.96e−06 [353.168]
	$W_2(1)$	3.01e−06 [13.203]	4.32e−06 [48.687]	5.70e−06 [119.068]	6.96e−06 [320.720]
Block component	$W_1(1)$	5.42e+03 [13.798]	8.50e+03 [51.361]	1.23e+04 [125.123]	1.52e+04 [355.182]
	$W_2(1)$	1.11e+04 [13.203]	1.74e+04 [49.265]	2.51e+04 [120.161]	3.13e+04 [349.141]
LU		1.05e−09 [127.660]	1.56e−09 [481.395]	Fail	Fail
Krawczyk		$K(U) \not\subset U$	$K(U) \not\subset U$	Fail	Fail
verifylss		3.46e−14 [29.672]	3.91e−14 [112.698]	Fail	Fail

The optimal condition for (4.8) is a saddle point problem

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -c \\ 0 \end{pmatrix},$$

where y is the Lagrange multiplier vector for the constraints $B^T x = 0$. In this problem, A is a positive semidefinite matrix and B has full-column rank.

We generate an original image of the cameraman as shown in Figure 2(a). The image is blurred by a Gaussian function

$$h(i,j) = e^{-(i^2+j^2)/18},$$

truncated such that the function has a support of 7×7, and then pixels are contaminated by Gaussian noise with the standard deviation of 0.05. The blurred and noisy image is shown in Figure 2(b). We solve the saddle point problem to find a restored image, which is shown in Figure 2(c). In the saddle point matrix, A has rank(A)=$2p - p_1 - p_2 + [p_1/2][p_2/2]$. Here $[\cdot]$ denotes the nearest integer.

Numerical results of the error bounds for the restored image are given in Table 5.

To end this section, we use a 3×3 block ill-conditioned saddle matrix to show that the error bound (2.5) is tight.

Example 4.5. We consider the following problem:

$$\mathcal{H} = \begin{pmatrix} \epsilon I & 0 & 0 \\ 0 & 0 & B \\ 0 & B^T & 0 \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix},$$

where $b_1, b_2, b_3 \in R^m, I, B \in R^{m\times m}$, and B is nonsingular. It is easy to find the inverse and the solution

$$\mathcal{H}^{-1} = \begin{pmatrix} \frac{1}{\epsilon} I & 0 & 0 \\ 0 & 0 & B^{-T} \\ 0 & B^{-1} & 0 \end{pmatrix}, \quad u^* = \begin{pmatrix} \frac{1}{\epsilon} b_1 \\ B^{-T} b_3 \\ B^{-1} b_2 \end{pmatrix}.$$

Consider $0 < \epsilon \leq 1$ and $\|B\|_\infty \geq 1$. The condition number of $\mathcal{H}$ satisfies

$$\|\mathcal{H}\|_\infty \left\|\mathcal{H}^{-1}\right\|_\infty \geq \epsilon^{-1} \|B\|_\infty.$$

When $\epsilon \to 0$, the condition number will go to ∞.

Using (2.2) or (2.5) with $W = B^{-1}B^{-T}$, we obtain

$$M(W) = \begin{pmatrix} \epsilon I & 0 \\ 0 & I \end{pmatrix}$$

and

$$\|u^* - u\|_2 \leq \frac{2}{\epsilon(\sqrt{5}-1)} \|b - \mathcal{H}u\|_2 \tag{4.9}$$

for $0 < \epsilon \leq 1/\|(B^TB)^{-1}\|_\infty$. Furthermore, the equality holds in (4.9) when $B = \frac{\epsilon(\sqrt{5}-1)}{2} I$ and $u = (\frac{1}{\epsilon} b_1, u_2, u_3)$.

5. Final remark. Using the algebraic analysis of a block diagonal preconditioner in [6], we proposed a fast verification method for saddle point linear systems where the $(1, 1)$ block may be singular. The method was implemented by using INTLAB [14] and taking all possible effects of rounding errors into account. Numerical results show that the method is efficient.

Acknowledgments. We are grateful to the two referees for their very helpful comments and suggestions. We wish to thank Prof. T. Yamamoto for carefully reading this paper and giving us helpful comments.

REFERENCES

[1] M. Benzi, G.H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[2] X. Chen and K. Hashimoto, *Numerical validation of solutions of saddle point matrix equations*, Numer. Linear Algebra Appl., 10 (2003), pp. 661–672.

[3] X. Chen and T. Kimura, *Finite element surface fitting for bridge management*, Internat. J. Inform. Technol. Decis. Mak., 5 (2006), pp. 671–681.

[4] *IEEE* 754-1985, *Standard for Binary Floating-Point Arithmetic*, IEEE, New York, 1985.

[5] H. Fu, M.K. Ng, M. Nikolova, and J.L. Barlow, *Efficient minimization methods of mixed $\ell 2-\ell 1$ and $\ell 1-\ell 1$ norms for image restoration*, SIAM J. Sci. Comput., 27 (2006), pp. 1881–1902.

[6] G.H. Golub, C. Greif, and J.M. Varah, *An algebraic analysis of a block diagonal preconditioner for saddle point systems*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 779–792.

[7] N.I.M. Gould, D. Orban, and P.L. Toint, *CUTEr, A Constrained and Unconstrained Testing Environment, Revisited*, http://hsl.rl.ac.uk/cuter-www/ (2002).
[8] S. Oishi and S.M. Rump, *Fast verification of solutions of matrix equations*, Numer. Math., 90 (2002), pp. 755–773.
[9] T. Ogita, S. Oishi, and Y. Ushiro, *Fast verification of solutions for sparse monotone matrix equations*, Comput. Suppl., 15 (2001), pp. 175–187.
[10] S. Roberts, M. Helgland, and I. Altas, *Approximation of a thin plate spline smoother using continuous piecewise polynomial functions*, SIAM J. Numer. Anal., 41 (2003), pp. 208–234.
[11] S.M. Rump, *Kleine Fehlerschranken bei Matrixproblemen*, Ph.D. thesis, Universität Karlsruhe, Karlsruhe, Germany, 1980.
[12] S.M. Rump, *Verification methods for dense and sparse system of equations*, in Topics in Validated Computation-Stud. Comput. Math., J. Herzberger, ed., Elsevier, Amsterdam, 1994, pp. 63–136.
[13] S.M. Rump, *Self-validating methods*, Linear Algebra Appl. 324 (2001), pp. 3–13.
[14] S.M. Rump, *Interval computations with INTLAB*, Braz. Electron. J. Math. Comput., 1 (1999), http://www.ti3.tu-harburg.de/publications/rump.
[15] A. Tikhonov and V. Arsenin, *Solution of Ill-Posed Problems*, Winston, Washington, D.C., 1977.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1709–1734

ON THE TENSOR SVD AND THE OPTIMAL LOW RANK ORTHOGONAL APPROXIMATION OF TENSORS*

JIE CHEN† AND YOUSEF SAAD†

Abstract. It is known that a higher order tensor does not necessarily have an optimal low rank approximation, and that a tensor might not be orthogonally decomposable (i.e., admit a tensor SVD). We provide several sufficient conditions which lead to the failure of the tensor SVD, and characterize the existence of the tensor SVD with respect to the higher order SVD (HOSVD). In the face of these difficulties to generalize standard results known in the matrix case to tensors, we consider the low rank orthogonal approximation of tensors. The existence of an optimal approximation is theoretically guaranteed under certain conditions, and this optimal approximation yields a tensor decomposition where the diagonal of the core is maximized. We present an algorithm to compute this approximation and analyze its convergence behavior. Numerical experiments indicate a linear convergence rate for this algorithm.

Key words. multilinear algebra, singular value decomposition, tensor decomposition, low rank approximation

AMS subject classifications. 15A69, 15A18

DOI. 10.1137/070711621

1. Introduction.

There has been renewed interest in studying the properties and decompositions of tensors (also known as N-way arrays or multidimensional arrays) in numerical linear algebra in recent years [30, 13, 12, 43, 17, 9, 28, 29, 15, 11]. The tensor approximation techniques have been fruitfully applied in various areas which include, among others, chemometrics [38, 4], signal processing [10, 8], vision and graphics [41, 42], and network analysis [31, 1]. From the point of view of practical applications, the matrix SVD and the optimal rank-r approximation of matrices (a.k.a. the Eckart–Young theorem [18]) are of particular interest, and it would be nice if these properties could be directly generalized to higher order tensors. However, for any order $N \geq 3$, de Silva and Lim [17] showed that the problem of optimal low rank approximation of higher order tensors is ill-posed for many ranks r, and that this ill-posedness is not rare for order-3 tensors. Furthermore, Kolda presented numerous examples to illustrate the difficulties of orthogonal tensor decompositions [28, 29]. These studies revealed many aspects of the dissimilarities between tensors and matrices, in spite of the fact that higher order tensors are multidimensional generalizations of matrices.

The most commonly used generalization of the matrix SVD to higher order tensors to date is the *higher order singular value decomposition* (HOSVD) [12]. The HOSVD decomposes an order-N tensor into a core tensor that has the same size as the original tensor together with N orthogonal[1] side-matrices. Although this decomposition preserves many nice aspects of the matrix SVD (e.g., the core has the all-orthogonality

*Received by the editors December 26, 2007; accepted for publication (in revised form) by L. De Lathauwer October 3, 2008; published electronically January 23, 2009. This work was supported by the NSF under grants DMS-0510131 and DMS-0528492 and by the Minnesota Supercomputing Institute.

http://www.siam.org/journals/simax/30-4/71162.html

†Department of Computer Science and Engineering, University of Minnesota at Twin Cities, Minneapolis, MN 55455 (jchen@cs.umn.edu, saad@cs.umn.edu).

[1]Throughout this paper, a matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$, is said to be *orthogonal* if $A^T A = I$. This generalizes the definition for square matrices.

property and the ordering property), a notable difference is that the core is in general not diagonal. Hence, in contrast with the matrix SVD, the HOSVD cannot be written as a sum of a few orthogonal outer-product terms.[2]

There exist three well-known approximations to higher order tensors: (1) the rank-1 approximation [13, 43, 27]; (2) the rank-$(r_1, r_2, \ldots, r_N)$ approximation with a full core and N orthogonal side-matrices (in the *Tucker/HOOI* fashion) [40, 13]; and (3) the approximation using r outer-product terms (in the *CANDECOMP/PARAFAC* fashion) [6, 19]. Note that the approximated tensor in case (3) might have rank less than r. Among these approximations, the rank-1 approximation [17] and the rank-$(r_1, r_2, \ldots, r_N)$ approximation are theoretically guaranteed to have a global optimum. In practical applications, the three approximations are generally computed using an alternating least squares (ALS) method [33, 3, 25] (the so-called workhorse algorithm [30]), although many other methods have also been proposed [34, 37, 43, 28, 15, 11]. The convergence behavior of the ALS method is theoretically unknown except under a few strong conditions [32]. Besides, it has long been observed that the ALS method for the PARAFAC model may converge extremely slowly if at all [36, 26]. An illustration of this phenomenon is given in the appendix.

Kolda [28] investigated several orthogonal decompositions of tensors related to different definitions of orthogonality, including *orthogonal rank decomposition*, *complete orthogonal rank decomposition*, and *strong orthogonal rank decomposition*. These decompositions might not be unique, or even exist. Among these definitions, only the *complete orthogonality* gives a situation which parallels that of the matrix SVD. This approach demands that the side-matrices all be orthogonal, in which case we use the term *tensor singular value decomposition* (tensor SVD; see Definition 4.1) in this paper. Zhang and Golub [43] proved that for all tensors of order $N \geq 3$, the tensor SVD is unique (up to signs) if it exists, and that the incremental rank-1 approximation approach will compute this decomposition.

The following contributions are made in this paper:

1. Sufficient conditions indicating which tensors fail to have a tensor SVD are given. These conditions are related to the rank, the order, and the dimensions of the tensor, and hence can be viewed as generalizations of results given in the literature with specific examples. Furthermore, the existence of the tensor SVD can be characterized by the diagonality of the core in the HOSVD of the tensor.
2. A form of low rank approximations—one that requires a diagonal core and orthogonal side-matrices—is discussed. Theoretically the global optimum of this approximation can be attained for any (appropriate) rank. We present an iterative algorithm to compute this approximation and analyze its convergence behavior.
3. The proposed approximation at the maximally possible rank leads to a decomposition of the tensor, where the diagonal of the core is maximized. This "maximal diagonality" for symmetric order-3 [14] and order-4 [7] tensors and for general order-3 tensors [15, 24, 35] has been previously investigated and Jacobi algorithms were used in the cited papers, but our discussion is in a more general context and the proposed algorithm is not of a Jacobi type.

2. Tensor algebra. In this section, we briefly review some concepts and notions that are used throughout the paper. A *tensor* is a multidimensional array of data whose elements are referred by using multiple indices. The number of indices required

[2]For discussions of orthogonality, see section 2.4.

is called the *order* of a tensor. We use

$$\mathcal{A} = (a_{i_1,i_2,\ldots,i_N}) \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$$

to denote a tensor $\mathcal{A}$ of order N. For $n = 1, 2, \ldots, N$, d_n is the nth *dimension* of $\mathcal{A}$. As a special case, a vector is an order-1 tensor and a matrix is an order-2 tensor.

2.1. Unfoldings and mode-n products. It is hard to visualize tensors of order $N > 3$. They can be flexibly represented when "unfolded" into matrices. The *unfolding* of a tensor along *mode* n is a matrix of dimension $d_n \times (d_{n+1} \cdots d_N d_1 \cdots d_{n-1})$. We denote the mode-$n$ unfolding of tensor $\mathcal{A}$ by $A_{(n)}$. Each column of $A_{(n)}$ is a column of $\mathcal{A}$ along the nth mode.

An important operation for a tensor is the *tensor-matrix multiplication*, also known as *mode-n product*. Given a tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ and a matrix $M \in \mathbb{R}^{c_n \times d_n}$, the mode-$n$ product is a tensor

$$\mathcal{B} = \mathcal{A} \times_n M \in \mathbb{R}^{d_1 \times \cdots \times d_{n-1} \times c_n \times d_{n+1} \cdots \times d_N},$$

where

$$b_{i_1,\ldots,i_{n-1},j_n,i_{n+1},\ldots,i_N} := \sum_{i_n=1}^{d_n} a_{i_1,\ldots,i_{n-1},i_n,i_{n+1},\ldots,i_N} m_{j_n,i_n}$$

for $j_n = 1, 2, \ldots, c_n$. In matrix representation, this is

$$B_{(n)} = M A_{(n)}. \tag{2.1}$$

2.2. Inner products and tensor norms. The *inner product* of two tensors $\mathcal{A}$ and $\mathcal{B}$ of the same size is defined by

$$\langle \mathcal{A}, \mathcal{B} \rangle_F := \sum_{i_N=1}^{d_N} \cdots \sum_{i_1=1}^{d_1} a_{i_1,\ldots,i_N} b_{i_1,\ldots,i_N},$$

and the *norm* induced from this inner product is

$$\|\mathcal{A}\|_F := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle_F}.$$

We say that $\mathcal{A}$ is a *unit tensor* if $\|\mathcal{A}\|_F = 1$. When $N = 2$, $\|\mathcal{A}\|_F$ is the Frobenius norm of matrix $\mathcal{A}$. The norm of a tensor is equal to the Frobenius norm of the unfolding of the tensor along any mode:

$$\|\mathcal{A}\|_F = \|A_{(n)}\|_F \quad \text{for } n = 1, \ldots, N.$$

2.3. Tensor products and outer products of vectors. The *tensor product* of an order-N tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ and an order-N' tensor $\mathcal{B} \in \mathbb{R}^{c_1 \times c_2 \times \cdots \times c_{N'}}$ is an order-$(N + N')$ tensor

$$\mathcal{C} = \mathcal{A} \otimes \mathcal{B} \in \mathbb{R}^{d_1 \times \cdots \times d_N \times c_1 \times \cdots \times c_{N'}},$$

where

$$c_{i_1,\ldots,i_N,j_1,\ldots,j_{N'}} := a_{i_1,\ldots,i_N} b_{j_1,\ldots,j_{N'}}.$$

Note that the operator $\otimes$ for tensor products unfortunately coincides with the one used to denote the Kronecker product of two matrices. In particular, the tensor product of two matrices (order-2 tensors) is an order-4 tensor, while the Kronecker product of two matrices is again a matrix. The reader shall not be confused by this notation since in this paper Kronecker products are not involved.

The *outer product* of N (column) vectors, which generalizes the standard outer product of two vectors (a rank-1 matrix), is a special case of tensor products. The outer product of N vectors $x^{(n)} \in \mathbb{R}^{d_n}$ is an order-N tensor of dimension $d_1 \times d_2 \times \cdots \times d_N$:

$$\mathcal{X} = x^{(1)} \otimes x^{(2)} \otimes \cdots \otimes x^{(N)}.$$

The $(i_1, i_2, \ldots, i_N)$-entry of $\mathcal{X}$ is $\prod_{n=1}^{N} (x^{(n)})_{i_n}$, where $(x^{(n)})_{i_n}$ denotes the i_nth entry of vector $x^{(n)}$. The tensor $\mathcal{X}$ is also called a *rank*-1 *tensor*. The *rank* of a tensor is defined in section 3.

It can be verified that the mode-n product of a rank-1 tensor $\mathcal{X}$ with a matrix M can be computed as follows:

$$\mathcal{X} \times_n M = x^{(1)} \otimes \cdots \otimes x^{(n-1)} \otimes \left(M x^{(n)}\right) \otimes x^{(n+1)} \cdots \otimes x^{(N)},$$

and it can be verified that the inner product of $\mathcal{X}$ with a general tensor $\mathcal{A}$ is

$$\begin{aligned} \langle \mathcal{A}, \mathcal{X} \rangle_F &= \left\langle \mathcal{A}, x^{(1)} \otimes x^{(2)} \otimes \cdots \otimes x^{(N)} \right\rangle_F \\ &= \mathcal{A} \times_1 x^{(1)^T} \times_2 x^{(2)^T} \times \cdots \times_N x^{(N)^T}. \end{aligned}$$

If $\mathcal{U} = u^{(1)} \otimes u^{(2)} \otimes \cdots \otimes u^{(N)}$ and $\mathcal{V} = v^{(1)} \otimes v^{(2)} \otimes \cdots \otimes v^{(N)}$ are two rank-1 tensors, then

$$\langle \mathcal{U}, \mathcal{V} \rangle_F = \prod_{n=1}^{N} \left\langle u^{(n)}, v^{(n)} \right\rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the standard Euclidean inner product of two vectors. A consequence of the above relation is that $\|\mathcal{U}\|_F$ is the product of the 2-norms of the vectors $u^{(n)}$'s.

2.4. Orthogonality of tensors. Two tensors $\mathcal{A}$ and $\mathcal{B}$ of the same size are *F-orthogonal* (*Frobenius orthogonal*) if their inner product is zero, i.e.,

$$\langle \mathcal{A}, \mathcal{B} \rangle_F = 0.$$

For rank-1 tensors $\mathcal{U} = u^{(1)} \otimes u^{(2)} \otimes \cdots \otimes u^{(N)}$ and $\mathcal{V} = v^{(1)} \otimes v^{(2)} \otimes \cdots \otimes v^{(N)}$, the above definition implies that they are F-orthogonal if

$$\prod_{n=1}^{N} \left\langle u^{(n)}, v^{(n)} \right\rangle = 0.$$

This leads to other options for defining orthogonality for two rank-1 tensors. The paper [28] discussed two cases:

1. Complete orthogonality: $\left\langle u^{(n)}, v^{(n)} \right\rangle = 0$ for all $n = 1, \ldots, N$.
2. Strong orthogonality: For all n, either $\left\langle u^{(n)}, v^{(n)} \right\rangle = 0$ or $u^{(n)}$ and $v^{(n)}$ are collinear, but there is at least one ℓ such that $\left\langle u^{(\ell)}, v^{(\ell)} \right\rangle = 0$.

In this paper we will simply use the term *orthogonal* for two outer products that are completely orthogonal (case 1).

2.5. Tensor decompositions. A *decomposition* of a tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ is of the form

$$\mathcal{A} = \mathcal{B} \times_1 S^{(1)} \times_2 S^{(2)} \times \cdots \times_N S^{(N)},$$

where $\mathcal{B} \in \mathbb{R}^{c_1 \times c_2 \times \cdots \times c_N}$ is called the *core tensor*, and $S^{(n)} \in \mathbb{R}^{d_n \times c_n}$ for $n = 1, \ldots, N$ are called *side-matrices*. An illustration is given in Figure 2.1.

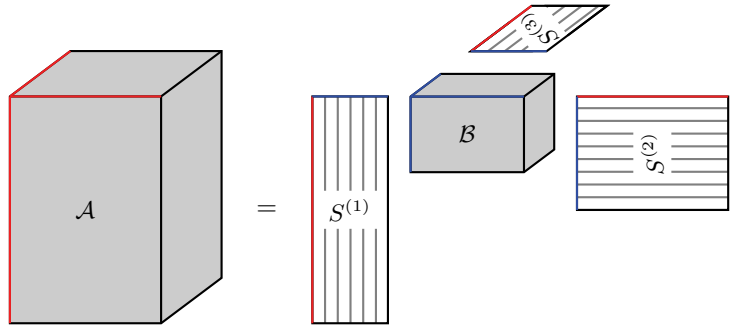


FIG. 2.1. *A decomposition of an order-3 tensor* $\mathcal{A}$ *as* $\mathcal{B} \times_1 S^{(1)} \times_2 S^{(2)} \times_3 S^{(3)}$.

Let $s_i^{(n)}$ be the ith column of $S^{(n)}$. The decomposition of $\mathcal{A}$ can equivalently be written as a linear combination of rank-1 tensors:

$$\mathcal{A} = \sum_{i_N=1}^{c_N} \cdots \sum_{i_1=1}^{c_1} b_{i_1,i_2,\ldots,i_N} s_{i_1}^{(1)} \otimes s_{i_2}^{(2)} \otimes \cdots \otimes s_{i_N}^{(N)}. \tag{2.2}$$

In particular, if $\mathcal{B}$ is diagonal, i.e., $b_{i_1,i_2,\ldots,i_N} = 0$ except when $i_1 = i_2 = \cdots = i_N$, then

$$\mathcal{A} = \sum_{i=1}^{r} b_{ii\ldots i} s_i^{(1)} \otimes s_i^{(2)} \otimes \cdots \otimes s_i^{(N)}, \tag{2.3}$$

where $r = \min\{c_1, \ldots, c_N\}$.

In the literature, the term "decomposition" is often used when "approximation" is meant instead. The *Tucker*3 *decomposition* is an approximation in the form of the right-hand side of (2.2), for given dimensions $c_1, c_2, \ldots, c_N$. Usually, it is required that c_n is less than the rank of $A_{(n)}$ for all n; otherwise the problem is trivial. The HOOI approach computes this approximation with an additional property that all the $S^{(n)}$'s are orthogonal matrices. The *CANDECOMP/PARAFAC decomposition* is an approximation in the form of the right-hand side of (2.3), for a prespecified r. Usually, r is smaller than the smallest dimension of all modes of $\mathcal{A}$, although requiring a larger r is also possible in the ALS and other algorithms. As will be discussed in the next section, the smallest r that satisfies equality (2.3) is the rank of the tensor $\mathcal{A}$.

3. Tensor ranks. The rank of a tensor causes difficulties when attempting to generalize matrix properties to tensors. There are several possible generalizations of the notion of rank. The *n-rank* of a tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$, for $n = 1, \ldots, N$, denoted by $\operatorname{rank}_n(\mathcal{A})$, is the rank of the unfolding $A_{(n)}$:

$$\operatorname{rank}_n(\mathcal{A}) := \operatorname{rank}(A_{(n)}).$$

The *(outer-product) rank* of $\mathcal{A}$, denoted by $\operatorname{rank}(\mathcal{A})$, is defined as

$$\operatorname{rank}(\mathcal{A}) := \min\left\{ r \,\middle|\, \exists\, x_i^{(n)} \in \mathbb{R}^{d_n},\ \ i = 1, \ldots, r,\ \ n = 1, \ldots, N, \right.$$
$$\left. \text{s.t. } \mathcal{A} = \sum_{i=1}^{r} x_i^{(1)} \otimes x_i^{(2)} \otimes \cdots \otimes x_i^{(N)} \right\}.$$

Hence, a tensor is the outer product of N vectors if and only if it has rank one, and the rank of a general tensor $\mathcal{A}$ is the minimum number of rank-1 tensors that sum to $\mathcal{A}$.

There are a few notable differences between the notion of rank for matrices and that for tensors:

1. For $N = 2$, i.e., when $\mathcal{A}$ is a matrix, $\operatorname{rank}_1(\mathcal{A})$ is the row rank, $\operatorname{rank}_2(\mathcal{A})$ is the column rank, and $\operatorname{rank}(\mathcal{A})$ is the outer-product rank, and they are all equal. However, for higher order tensors ($N > 2$), in general, the n-ranks are different for different modes n, and they are different from $\operatorname{rank}(\mathcal{A})$ [12]. Furthermore, the rank of a matrix A cannot be larger than the smallest dimension of both modes of A, but for tensors this is no longer true; i.e., the rank can be larger than the smallest dimension of the tensor [12].

2. The matrix SVD yields one possible way of writing a matrix as a sum of outer-product terms, and the number of nonzero singular values is equal to the rank of the matrix. However, a tensor SVD does not always exist (see section 4), but if it indeed does, it is unique up to signs [34, 43] and the number of singular values is equal to the rank of the tensor (see Definition 4.1 and Proposition 4.2).

3. It is well known that the optimal rank-r approximation of a matrix is simply its truncated SVD. However, some tensors may fail to have an optimal rank-r approximation [17]. If such an approximation exists, then it is unclear whether it can be written in the form of a diagonal core multiplied by orthogonal side-matrices.

Next are some lemmas and a theorem related to tensor ranks, which were also given in [17]. They are useful in deriving the results in section 4. The first lemma indicates that the rank of a tensor cannot be smaller than any of its n-ranks.

LEMMA 3.1. *Let $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ be an order-N tensor. Then*

$$\operatorname{rank}_n(\mathcal{A}) \leq \min\{\operatorname{rank}(\mathcal{A}), d_n\} \quad \textit{for } n = 1, 2, \ldots, N.$$

The next lemma illustrates a way to construct higher order tensors while preserving the rank.

LEMMA 3.2. *Let $\mathcal{A}$ be a tensor and x be a nonzero vector. Then*

$$\operatorname{rank}(\mathcal{A}) = \operatorname{rank}(\mathcal{A} \otimes x).$$

The following lemma indicates that given any dimension $d_1 \times d_2 \times \cdots \times d_N$, we can construct a tensor of arbitrary rank $R \leq \min\{d_1, d_2, \ldots, d_N\}$.

LEMMA 3.3. *For $n = 1, \ldots, N$, let $x_1^{(n)}, \ldots, x_R^{(n)} \in \mathbb{R}^{d_n}$ be linearly independent. Then the tensor*

$$\mathcal{A} = \sum_{i=1}^{R} x_i^{(1)} \otimes x_i^{(2)} \otimes \cdots \otimes x_i^{(N)}$$

has rank R.

The next theorem is due to JáJá and Takche [23]. They showed that if $\mathcal{A}$ and $\mathcal{B}$ are order-3 tensors and at least one of them is a "stack" of two matrices, then the rank of their direct sum is equal to the sum of their ranks.

THEOREM 3.4 (JáJá–Takche). *Let* $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ *and* $\mathcal{B} \in \mathbb{R}^{c_1 \times c_2 \times c_3}$. *If* $2 \in \{d_1, d_2, d_3, c_1, c_2, c_3\}$, *then*

$$\operatorname{rank}(\mathcal{A} \oplus \mathcal{B}) = \operatorname{rank}(\mathcal{A}) + \operatorname{rank}(\mathcal{B}).$$

3.1. The ill-posedness of the optimal low rank approximation problem. de Silva and Lim [17] proved that for any order $N \geq 3$ and dimensions $d_1, \ldots, d_N \geq 2$, there exists a rank-$(r+1)$ tensor that has no optimal rank-r approximation, for any $r = 2, \ldots, \min\{d_1, \ldots, d_N\}$. This result was further generalized to an arbitrary rank gap, i.e., there exists a rank-$(r+s)$ tensor that has no optimal rank-r approximation, for some r's and s's.

Essentially, this ill-posedness of the optimal approximation problem is illustrated by the fact that the tensor

$$\mathcal{E} := e_2 \otimes e_1 \otimes e_1 + e_1 \otimes e_2 \otimes e_1 + e_1 \otimes e_1 \otimes e_2 \in \mathbb{R}^{2\times 2\times 2},$$

where e_i is the ith column of the identity matrix, has rank 3 but can be approximated arbitrarily closely by rank-at-most-2 tensors. Hence $\mathcal{E}$ does not have an optimal rank-2 approximation. Then according to Theorem 3.4 and Lemma 3.2, the ill-posedness of the problem can be generalized to arbitrary rank and order, by constructing higher rank and higher order tensors using direct sums and tensor products. We restate one of the results of [17] in the following theorem. For details of the proof, see the original paper.

THEOREM 3.5. *For* $N \geq 3$ *and* $d_1, d_2, \ldots, d_N \geq 2$, *there exists a tensor* $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ *of rank* $r+s$ *that has no optimal rank-*r *approximation, for any* r *and* $s \geq 1$ *satisfying* $2s \leq r \leq \min\{d_1, d_2, \ldots, d_N\}$.

4. The tensor SVD and its (non-)existence. The definition used for the SVD of a tensor generalizes the matrix SVD from the angle of an expansion of outer product matrices, which becomes an expansion into a sum of rank-1 tensors.

DEFINITION 4.1. *If a tensor* $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ *can be written in the form*

$$\mathcal{A} = \sum_{i=1}^{R} \sigma_i u_i^{(1)} \otimes u_i^{(2)} \otimes \cdots \otimes u_i^{(N)}, \tag{4.1}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R > 0$ *and* $\langle u_j^{(n)} u_k^{(n)} \rangle = \delta_{jk}$ *(Kronecker delta) for* $n = 1, 2, \ldots, N$, *then* (4.1) *is said to be the* tensor singular value decomposition (tensor SVD) *of* $\mathcal{A}$. *The* σ_i*'s are* singular values *and the* $u_i^{(n)}$*'s for* $i = 1, \ldots, R$ *are the mode-*n singular vectors.

We also call (4.1) the *SVD* of tensor $\mathcal{A}$ for short where there is no ambiguity about tensors and matrices. In fact, when $N = 2$, i.e., $\mathcal{A}$ is a matrix, the tensor SVD of $\mathcal{A}$ boils down to the matrix SVD. Expression (4.1) can equivalently be written in the form

$$\mathcal{A} = \mathcal{D} \times_1 U^{(1)} \times_2 U^{(2)} \times \cdots \times_N U^{(N)}, \tag{4.2}$$

where $\mathcal{D} \in \mathbb{R}^{R \times R \times \cdots \times R}$ is the diagonal *core tensor* with $\mathcal{D}_{ii\ldots i} = \sigma_i$, and

$$U^{(n)} = \left[u_1^{(n)}, u_2^{(n)}, \ldots, u_R^{(n)} \right] \in \mathbb{R}^{d_n \times R} \tag{4.3}$$

are orthogonal matrices for $n = 1, 2, \ldots, N$. The following proposition indicates that the tensor SVD is rank revealing.

PROPOSITION 4.2. *If a tensor $\mathcal{A}$ has the SVD as* (4.1)*, then* $\operatorname{rank}(\mathcal{A}) = R$.

Proof. This follows from Lemma 3.3. ∎

Trivially, if a tensor is constructed as in (4.1), then its SVD exists. However, in general, a tensor of order $N \geq 3$ may fail to have such a decomposition. In this section, we identify some of these situations.

To begin with, note that the orthogonality of each $U^{(n)}$ implies that $R \leq d_n$ for each n, i.e., $R \leq \min\{d_1, d_2, \ldots, d_N\}$. This leads to the following simple result.

PROPOSITION 4.3. *A tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ with* $\operatorname{rank}(\mathcal{A}) > \min\{d_1, d_2, \ldots, d_N\}$ *does not admit a tensor SVD.*

Proof. The existence of a tensor SVD such as in (4.1) would trivially lead to a contradiction since the tensor in (4.1) has rank R with $R \leq \min\{d_1, d_2, \ldots, d_N\}$. ∎

Note that Theorem 3.5 guarantees that the condition of Proposition 4.3 is not vacuously satisfied, for any order $N \geq 3$ and dimensions $d_1, d_2, \ldots, d_N \geq 2$.

COROLLARY 4.4. *Given a tensor $\mathcal{A}$ satisfying the condition in Proposition* 4.3*, any tensor of the form*

$$\mathcal{A} \otimes x^{(N+1)} \otimes \cdots \otimes x^{(N+\ell)},$$

where $\ell \geq 1$ and $x^{(N+1)}, \ldots, x^{(N+\ell)}$ are nonzero vectors, does not admit a tensor SVD.

Proof. This follows from Proposition 4.3 and Lemma 3.2. ∎

COROLLARY 4.5. *A tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ does not admit a tensor SVD if there exists at least one mode n such that* $\operatorname{rank}_n(\mathcal{A}) > \min\{d_1, d_2, \ldots, d_N\}$.

Proof. This follows from Proposition 4.3 and Lemma 3.1. ∎

PROPOSITION 4.6. *There exists a tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ which does not admit a tensor SVD whenever*

$$d := \max_n\{d_n\} > \min_n\{d_n\} \quad \text{and} \quad d^2 \leq \prod_{n=1}^{N} d_n.$$

Proof. Without loss of generality, assume that $d = d_1 \geq d_2 \geq \cdots \geq d_N$ and let $d' = d_2 \times \cdots \times d_N$. Since $d \leq d'$, for an arbitrary set of orthonormal vectors $\{a_i \in \mathbb{R}^{d'} \mid i = 1, \ldots, d\}$, we can construct a tensor $\mathcal{A}$ whose unfolding $A_{(1)} = [a_1, a_2, \ldots, a_d]^T$. Then $\operatorname{rank}_1(\mathcal{A}) = d$. By Corollary 4.5, $\mathcal{A}$ does not admit a tensor SVD. ∎

Note that when $N = 2$, i.e., for the matrix case, it is impossible for d_1 and d_2 to satisfy the condition in the proposition.

In closing this section, we provide a necessary and sufficient condition to characterize the existence of the tensor SVD.[3] This is related to the HOSVD proposed by [12]. The essential relation underlying the theorem is that the mode-n singular vectors of $\mathcal{A}$, when its SVD exists, are also the left singular vectors of the unfolding $A_{(n)}$.

THEOREM 4.7. *A tensor $\mathcal{A}$ admits an SVD if and only if there exists an HOSVD of $\mathcal{A}$ such that the core is diagonal.*

Proof. The sufficient condition is obvious. Consider the necessary condition. If $\mathcal{A}$ can be written in the form (4.1), then define the tensor

$$\mathcal{W}_i^{(n)} := u_i^{(n+1)} \otimes \cdots \otimes u_i^{(N)} \otimes u_i^{(1)} \otimes \cdots \otimes u_i^{(n-1)},$$

[3] As pointed out by a referee, the provided relation may have long been recognized in other fields of research, such as signal processing, at least in the case of distinct singular values.

and let $w_i^{(n)}$ be the vectorization of $\mathcal{W}_i^{(n)}$. Then the unfolding of $\mathcal{A}$ along mode n is

$$A_{(n)} = \sum_{i=1}^{R} \sigma_i u_i^{(n)} \otimes w_i^{(n)}.$$

Since $\langle u_j^{(n)} u_k^{(n)} \rangle = \delta_{jk}$ for all n, we have $\langle w_j^{(n)} w_k^{(n)} \rangle = \delta_{jk}$. Hence the above form is the SVD of matrix $A_{(n)}$. In other words, the vectors $u_1^{(n)}, \ldots, u_R^{(n)}$ are the left singular vectors of $A_{(n)}$. From the construction of the HOSVD, (4.2) is a valid HOSVD for $\mathcal{A}$.[4] □

The proof of the above theorem indicates that if the SVD of a tensor $\mathcal{A}$ exists, then its singular values coincide with the nonzero mode-n singular values in its HOSVD. However, the HOSVD of a tensor may not be unique, since the SVD of the unfoldings $A_{(n)}$'s are not guaranteed to be unique. Hence even if a tensor is constructed as in (4.1), its HOSVD will not necessarily recover this form. This is the reason why in the above theorem we use the phase "... if there exists ...."

It is interesting to note again that the nonuniqueness of matrix SVD is caused by duplicate singular values; however, the tensor SVD is unique (if it exists) even when some of the singular values are the same [43, Theorem 3.2].

5. The optimal low rank orthogonal approximation. The problem addressed by tensor analysis is to approximate some tensor $\mathcal{A}$ by a linear combination of tensors $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_r$ that have "special" structures, e.g., rank-1 tensors, orthogonal tensors, or simple tensors.[5] For this it is desirable to minimize

$$\left\| \mathcal{A} - \sum_{i=1}^{r} \sigma_i \mathcal{T}_i \right\|_F$$

for a given r. Without loss of generality, we assume that $\|\mathcal{T}_i\|_F = 1$ for all i. As discussed in section 3.1, if the $\mathcal{T}_i$'s are rank-1 tensors, then the infimum of the above expression might not necessarily be attained. The following proposition reveals some properties when the infimum is indeed achieved.

PROPOSITION 5.1. *Given a tensor $\mathcal{A}$ and a positive integer r, consider a set of linear combinations of tensors of the form*

$$\mathcal{T} := \sum_{i=1}^{r} \sigma_i \mathcal{T}_i, \tag{5.1}$$

where the $\mathcal{T}_i$'s are arbitrary unit tensors. If $\inf \|\mathcal{A} - \mathcal{T}\|_F$ is reached on this set, then for the optimal $\mathcal{T}$ and $\mathcal{T}_i$'s,

$$\langle \mathcal{A} - \mathcal{T}, \mathcal{T}_i \rangle_F = 0 \quad \text{for } i = 1, 2, \ldots, r.$$

Furthermore, if the $\mathcal{T}_i$'s are required to be mutually F-orthogonal, then the optimal σ_i's are related to the optimal $\mathcal{T}_i$'s by

$$\sigma_i = \langle \mathcal{A}, \mathcal{T}_i \rangle_F \quad \text{for } i = 1, 2, \ldots, r. \tag{5.2}$$

[4] In order to strictly conform to the definition of the HOSVD defined in [12], in (4.2) the size of $\mathcal{D}$ should be enlarged from $R \times R \times \cdots \times R$ to $d_1 \times d_2 \times \cdots \times d_N$ by padding zeros, and the $U^{(n)}$'s should be padded with orthogonal columns to make square shapes.

[5] A tensor is simple if it is the tensor product of two tensors.

In this situation,

$$\|\mathcal{T}\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i^2} \qquad \textit{and} \qquad \|\mathcal{A} - \mathcal{T}\|_F^2 = \|\mathcal{A}\|_F^2 - \|\mathcal{T}\|_F^2 . \tag{5.3}$$

Proof. If the infimum is attained by a certain set of σ_i's and $\mathcal{T}_i$'s, and if there is a j such that $\langle \mathcal{A} - \mathcal{T}, \mathcal{T}_j \rangle_F = \epsilon \neq 0$, then

$$\begin{aligned}
&\left\| \mathcal{A} - \sum_{i=1}^{r} \sigma_i \mathcal{T}_i - \epsilon \mathcal{T}_j \right\|_F^2 \\
&= \left\| \mathcal{A} - \sum_{i=1}^{r} \sigma_i \mathcal{T}_i \right\|_F^2 - 2\epsilon \left\langle \mathcal{A} - \sum_{i=1}^{r} \sigma_i \mathcal{T}_i, \mathcal{T}_j \right\rangle_F + \epsilon^2 \|\mathcal{T}_j\|_F^2 \\
&= \left\| \mathcal{A} - \sum_{i=1}^{r} \sigma_i \mathcal{T}_i \right\|_F^2 - \epsilon^2 < \left\| \mathcal{A} - \sum_{i=1}^{r} \sigma_i \mathcal{T}_i \right\|_F^2 ,
\end{aligned}$$

which contradicts the assumption.

If the unit tensors $\mathcal{T}_i$'s are mutually F-orthogonal, then

$$0 = \left\langle \mathcal{A} - \sum_{i=1}^{r} \sigma_i \mathcal{T}_i, \mathcal{T}_j \right\rangle_F = \langle \mathcal{A}, \mathcal{T}_j \rangle_F - \sigma_j \langle \mathcal{T}_j, \mathcal{T}_j \rangle_F = \langle \mathcal{A}, \mathcal{T}_j \rangle_F - \sigma_j .$$

Also,

$$\|\mathcal{T}\|_F^2 = \langle \mathcal{T}, \mathcal{T} \rangle_F = \sum_{i,j=1}^{r} \langle \sigma_i \mathcal{T}_i, \sigma_j \mathcal{T}_j \rangle_F = \sum_{i=1}^{r} \sigma_i^2$$

and

$$\begin{aligned}
\left\| \mathcal{A} - \sum_{i=1}^{r} \sigma_i \mathcal{T}_i \right\|_F^2 &= \|\mathcal{A}\|_F^2 - \sum_{i=1}^{r} 2\sigma_i \langle \mathcal{A}, \mathcal{T}_i \rangle_F + \sum_{i=1}^{r} \sigma_i^2 \\
&= \|\mathcal{A}\|_F^2 - \sum_{i=1}^{r} \sigma_i^2 = \|\mathcal{A}\|_F^2 - \|\mathcal{T}\|_F^2 . \qquad \square
\end{aligned}$$

The last part of the proof indicates that the equalities in (5.3) follow from the orthogonality of the $\mathcal{T}_i$'s and the relations (5.2). They do not require optimality.

In this section, we will see that if the $\mathcal{T}_i$'s are mutually orthogonal rank-1 tensors, then the infimum in the proposition can be attained. Formally, we will prove that the problem

$$\begin{aligned}
&\min \quad E := \left\| \mathcal{A} - \sum_{i=1}^{r} \sigma_i u_i^{(1)} \otimes u_i^{(2)} \otimes \cdots \otimes u_i^{(N)} \right\|_F \\
&\text{s.t.} \quad \left\langle u_j^{(n)}, u_k^{(n)} \right\rangle = \delta_{jk}, \quad \text{for } n = 1, 2, \ldots, N,
\end{aligned} \tag{5.4}$$

has a solution for any $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ and any $r \leq \min\{d_1, d_2, \ldots, d_N\}$. The solution for the case $r = \min\{d_1, d_2, \ldots, d_N\}$ leads to a decomposition of $\mathcal{A}$ where the diagonal of the core is maximized.

5.1. Existence of the global optimum. Let

$$\mathcal{T}_i := u_i^{(1)} \otimes u_i^{(2)} \otimes \cdots \otimes u_i^{(N)} \quad \text{for } i = 1, \ldots, r, \tag{5.5}$$

and let σ_i's be defined as in (5.2); then, according to Proposition 5.1 (see comments following the proof),

$$E^2 = \left\| \mathcal{A} - \sum_{i=1}^{r} \sigma_i \mathcal{T}_i \right\|_F^2 = \|\mathcal{A}\|_F^2 - \sum_{i=1}^{r} \sigma_i^2.$$

Hence minimizing E is equivalent to maximizing $\sum_{i=1}^{r} \sigma_i^2$; i.e., the optimization problem (5.4) is equivalent to the following:

$$\begin{aligned} \max \quad & E' := \sum_{i=1}^{r} \left(\mathcal{A} \times_1 u_i^{(1)^T} \times_2 u_i^{(2)^T} \times \cdots \times_N u_i^{(N)^T} \right)^2 \\ \text{s.t.} \quad & \left\langle u_j^{(n)}, u_k^{(n)} \right\rangle = \delta_{jk} \quad \text{for } n = 1, 2, \ldots, N. \end{aligned} \tag{5.6}$$

Let

$$U^{(n)} = \left[u_1^{(n)}, u_2^{(n)}, \ldots, u_r^{(n)} \right] \in \Omega^{(n)}, \tag{5.7}$$

where

$$\Omega^{(n)} := \{ W \in \mathbb{R}^{d_n \times r} \mid W^T W = I \} \tag{5.8}$$

for $n = 1, 2, \ldots, N$. The problem (5.6) can be interpreted as that of maximizing E' within the feasible region

$$\Omega := \Omega^{(1)} \times \Omega^{(2)} \times \cdots \times \Omega^{(N)}. \tag{5.9}$$

Since for each n the set $\Omega^{(n)}$ is compact (see, e.g., [22, p. 69]), by the Tychonoff theorem, the feasible region Ω is compact. Under the continuous mapping E', the image $E'(\Omega)$ is also compact. Hence it has a maximum. This proves the following theorem.

THEOREM 5.2. *There exists a solution to the problem* (5.6) *(or equivalently* (5.4) *with σ_i defined in* (5.2)*) for any $r \leq \min\{d_1, d_2, \ldots, d_N\}$.*

5.2. Relation to tensor decompositions. Let $U^{(n)}$, $n = 1, \ldots, N$, be the solution to the problem (5.4) with $r = \min\{d_1, d_2, \ldots, d_N\}$ and σ_i be defined in (5.2). Also for $n = 1, \ldots, N$, let $U^{(n)^\perp}$ be a $d_n \times (d_n - r)$ matrix such that the square matrix

$$\tilde{U}^{(n)} := \left[U^{(n)}, U^{(n)^\perp} \right] \in \mathbb{R}^{d_n \times d_n} \tag{5.10}$$

is orthogonal. Further, define the tensor

$$\mathcal{S} := \mathcal{A} \times_1 \tilde{U}^{(1)^T} \times_2 \tilde{U}^{(2)^T} \times \cdots \times_N \tilde{U}^{(N)^T} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}. \tag{5.11}$$

Then the equality

$$\mathcal{A} = \mathcal{S} \times_1 \tilde{U}^{(1)} \times_2 \tilde{U}^{(2)} \times \cdots \times_N \tilde{U}^{(N)} \tag{5.12}$$

holds. This decomposition of $\mathcal{A}$ has the following two properties:

(i) The side-matrices $\tilde{U}^{(n)}$'s are orthogonal.
(ii) The (squared) norm of the diagonal of the core $\mathcal{S}$

$$\sum_{i=1}^{\min\{d_1,\ldots,d_N\}} s_{ii\ldots i}^2 = \sum_{i=1}^{r} \left(\mathcal{A} \times_1 u_i^{(1)^T} \times_2 u_i^{(2)^T} \times \cdots \times_N u_i^{(N)^T} \right)^2 = \sum_{i=1}^{r} \sigma_i^2$$

is maximized among all choices of the orthogonal side-matrices. This is known as *maximal diagonality* in [12].

5.3. First order condition. The Lagrangian of (5.6) is

$$L = \sum_{i=1}^{r} \sigma_i^2 - \sum_{j,k=1}^{r} \sum_{n=1}^{N} \mu_{j,k}^n \left(\left\langle u_j^{(n)}, u_k^{(n)} \right\rangle - \delta_{jk} \right), \tag{5.13}$$

where

$$\sigma_i = \mathcal{A} \times_1 u_i^{(1)^T} \times_2 u_i^{(2)^T} \times \cdots \times_N u_i^{(N)^T} \tag{5.14}$$

and the $\mu_{j,k}^n$'s are Lagrange multipliers. Define the vector

$$\begin{aligned} v_i^{(n)} &:= \mathcal{A} \times_1 u_i^{(1)^T} \times \cdots \times_{n-1} u_i^{(n-1)^T} \times_{n+1} u_i^{(n+1)^T} \times \cdots \times_N u_i^{(N)^T} \\ &\in \mathbb{R}^{1\times\cdots\times 1\times d_n\times 1\cdots\times 1}. \end{aligned} \tag{5.15}$$

(Here we abuse the use of notation "=." More precisely, $v_i^{(n)}$ should be the mode-n unfolding of the tensor on the right-hand side of (5.15).) It is not hard to see that

$$\left\langle u_i^{(n)}, v_i^{(n)} \right\rangle = \sigma_i$$

for all n and i, and $v_i^{(n)}$ is the partial derivative of σ_i with respect to $u_i^{(n)}$.

The partial derivative of the Lagrangian with respect to $u_i^{(n)}$ is

$$\frac{\partial L}{\partial u_i^{(n)}} = 2\sigma_i v_i^{(n)} - \sum_{j=1}^{r} \mu_{j,i}^n u_j^{(n)} - \sum_{k=1}^{r} \mu_{i,k}^n u_k^{(n)}$$

for any n and i. By setting the partial derivatives to 0 and putting all equations related to the same n in matrix form, we obtain the following equations:

$$\begin{bmatrix} v_1^{(n)} & \cdots & v_r^{(n)} \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} = \begin{bmatrix} u_1^{(n)} & \cdots & u_r^{(n)} \end{bmatrix} \begin{bmatrix} \frac{\mu_{1,1}^n + \mu_{1,1}^n}{2} & \cdots & \frac{\mu_{1,r}^n + \mu_{r,1}^n}{2} \\ \vdots & \ddots & \vdots \\ \frac{\mu_{r,1}^n + \mu_{1,r}^n}{2} & \cdots & \frac{\mu_{r,r}^n + \mu_{r,r}^n}{2} \end{bmatrix} \tag{5.16}$$

for all $n = 1, 2, \ldots, N$. Let

$$V^{(n)} := \left[v_1^{(n)}, v_2^{(n)}, \ldots, v_r^{(n)} \right], \tag{5.17}$$

$$\Sigma := \operatorname{diag}(\sigma_1, \ldots, \sigma_r), \tag{5.18}$$

and let $M^{(n)}$ be the second term on the right-hand side of (5.16). Then (5.16) is compactly represented as

$$V^{(n)}\Sigma = U^{(n)}M^{(n)}, \quad n = 1, 2, \ldots, N. \tag{5.19}$$

In summary, the necessary condition of an extremum of the Lagrangian is (5.19), where $V^{(n)}$ is defined in (5.17), Σ is defined in (5.18), $U^{(n)}$ is defined in (5.7), and $M^{(n)}$ is symmetric, for all $n = 1, 2, \ldots, N$.

5.4. Algorithm: LROAT. We seek orthogonal matrices $U^{(n)}$'s and symmetric matrices $M^{(n)}$'s which satisfy the system (5.19). (The Σ and $V^{(n)}$'s are computed from the $U^{(n)}$'s.) Note that the pair $U^{(n)}, M^{(n)}$ happens to be the polar decomposition of the matrix $V^{(n)}\Sigma$. Hence the system can be solved in an iterative fashion: We begin with an initial guess of the set of orthogonal matrices $\{U^{(1)}, U^{(2)}, \ldots, U^{(N)}\}$, which can be obtained, say, by the HOSVD of $\mathcal{A}$. For each n, we compute $V^{(n)}$ and Σ, and update $U^{(n)}$ as an orthogonal polar factor of $V^{(n)}\Sigma$. This procedure is iterated until convergence is observed. Algorithm 1 (LROAT) summarizes this idea.

ALGORITHM 1. Low Rank Orthogonal Approximation of Tensors (LROAT)

Input: Tensor $\mathcal{A}$, rank r, orthogonal matrices $U^{(1)}, \ldots, U^{(N)}$ as initial guess
Output: $\sigma_1, \ldots, \sigma_r$, $U^{(1)}, \ldots, U^{(N)}$

```
repeat
  for n ← 1, ..., N do
    Compute V^(n) = [v_1^(n), ..., v_r^(n)] according to (5.15)
    Compute Σ = diag(σ_1, ..., σ_r) according to (5.14)
    [Q^(n), H^(n)] ← polar-decomp(V^(n)Σ)
    Update U^(n) ← Q^(n)
  end for
until convergence
```

Note that when $r = 1$, the matrix $V^{(n)} = [v_1^{(n)}]$ and $U^{(n)} = [u_1^{(n)}]$, which means that for each iteration $u_1^{(n)}$ is updated as the normalized $v_1^{(n)}$. This indicates that the LROAT algorithm for $r = 1$ boils down to the ALS method [43] (or the so-called higher order power method [13, 27]) for computing the optimal rank-1 approximation. Hence, it is not unexpected to see in the numerical experiments that in general LROAT converges linearly. We also comment that LROAT is not an ALS method (except for the case $r = 1$) by the nature of the update of $U^{(n)}$.

5.5. Convergence analysis. LROAT employs an alternating procedure (iterating through $U^{(1)}$, $U^{(2)}, \ldots,$ $U^{(N)}$), where in each step all but one of the $(U^{(n)})$ parameters are fixed. In general, algorithms of this type, including ALS, are not guaranteed to converge. Specifically, the objective function may converge but not the parameters. (See, for example, [32] for some discussions.) For LROAT, we are also unable yet to prove the global convergence, though empirically it appears to hold. However, in this section, we will prove the following: (1) the iterations monotonically increase the objective value E' (Theorem 5.4); (2) under a mild condition, of the generated parameter sequence, every converging subsequence converges to a stationary point of the objective function (Theorem 5.7); and (3) in a neighborhood of a local maximum, the parameter sequence converges to this stationary point (Theorem 5.9).

Before analyzing the convergence behavior of LROAT, we index all of the iterates. The outer loop is indexed by p and the overall iterations are indexed by idx, which is equal to $n+(p-1)N$. In other words, Algorithm 1 is rewritten as follows. In particular, the numbered lines correspond to the lines in Algorithm 1.

---

for $p \leftarrow 1, 2, \ldots$ **do**
  for $n \leftarrow 1, \ldots, N$ **do**
    $idx = n + (p-1)N$
    For all i, compute $\sigma_i^{(idx)}$ according to $U_{(p+1)}^{(1)}, \ldots, U_{(p+1)}^{(n-1)}, U_{(p)}^{(n)}, U_{(p)}^{(n+1)}, \ldots, U_{(p)}^{(N)}$
    Objective $E'^{(idx)} = \sum_{i=1}^{r} \left(\sigma_i^{(idx)}\right)^2$
3: Compute $V_{(p)}^{(n)}$ from $U_{(p+1)}^{(1)}, \ldots, U_{(p+1)}^{(n-1)}, U_{(p)}^{(n+1)}, \ldots, U_{(p)}^{(N)}$
4: Assign $\Sigma^{(idx)} = \operatorname{diag}\left(\sigma_1^{(idx)}, \ldots, \sigma_r^{(idx)}\right)$
5: Polar decomposition $V_{(p)}^{(n)}\Sigma^{(idx)} = Q_{(p)}^{(n)}H_{(p)}^{(n)}$
6: Update $U_{(p+1)}^{(n)} = Q_{(p)}^{(n)}$
  end for
end for

The following lemma, which is well known when the matrix A is square, reveals the trace maximizing property that is important for the convergence analysis of LROAT.

LEMMA 5.3. *Let matrix $A \in \mathbb{R}^{m\times n}$, $m \geq n$, have the polar decomposition $A = QH$, where $Q \in \mathbb{R}^{m\times n}$ is the orthogonal polar factor and $H \in \mathbb{R}^{n\times n}$ is the symmetric positive semidefinite factor; then*

$$\max_{P\in\mathbb{R}^{m\times n},\, P^TP=I} \operatorname{tr}(P^TA)$$

is attained when $P = Q$.

Proof. Any P can be written as ZQ, where $Z \in \mathbb{R}^{m\times m}$ is orthogonal. Then

$$\operatorname{tr}(P^TA) = \operatorname{tr}(Q^TZ^TQH) = \operatorname{tr}(Z^TQHQ^T).$$

Since QHQ^T is symmetric positive semidefinite, $\max \operatorname{tr}(Z^TQHQ^T)$ is attained when $Z = I$. ∎

Since $U_{(p+1)}^{(n)}$ is the orthogonal polar factor of $V_{(p)}^{(n)}\Sigma^{(idx)}$, by Lemma 5.3,

$$\sum_{i=1}^{r}\left(\sigma_i^{(idx)}\right)^2 = \operatorname{tr}\left(U_{(p)}^{(n)^T}V_{(p)}^{(n)}\Sigma^{(idx)}\right) \leq \operatorname{tr}\left(U_{(p+1)}^{(n)^T}V_{(p)}^{(n)}\Sigma^{(idx)}\right) = \sum_{i=1}^{r}\sigma_i^{(idx+1)}\sigma_i^{(idx)}.$$

Then, by the Cauchy–Schwarz inequality,

$$\sum_{i=1}^{r}\left(\sigma_i^{(idx)}\right)^2 \leq \sum_{i=1}^{r}\sigma_i^{(idx+1)}\sigma_i^{(idx)} \leq \sum_{i=1}^{r}\left(\sigma_i^{(idx+1)}\right)^2, \tag{5.20}$$

and

$$\sum_{i=1}^{r}\left(\sigma_i^{(idx)}\right)^2 = \sum_{i=1}^{r}\left(\sigma_i^{(idx+1)}\right)^2 \quad \text{iff} \quad \sigma_i^{(idx)} = \sigma_i^{(idx+1)} \text{ for all } i. \tag{5.21}$$

Inequality (5.20) means that each update of $U^{(n)}$ increases the value of the objective function E', i.e.,

$$E'^{(idx)} \leq E'^{(idx+1)}.$$

Since E' is bounded from above (existence of the maximum; see Theorem 5.2), the sequence $\{E'^{(idx)}\}_{idx=1}^{\infty}$ converges. Note that the convergence does not depend on the initial guess input to the algorithm. Formally, we have established the following theorem.

THEOREM 5.4. *Given any initial guess, the iterations of Algorithm* 1 *monotonically increase the objective function E' defined in* (5.6) *to a limit.*

The convergence of the objective function does not necessarily imply that the function parameters will converge. However, in our case, since the parameters $U^{(n)}$'s are bounded, they admit converging subsequences. Next we will show that every such subsequence converges to a stationary point of E'. For this, the following lemma uses a helper function f.

LEMMA 5.5. *Let $T : \Theta \to \Theta$ be a continuous mapping and a sequence $\{\theta_n \in \Theta\}_{n=1}^{\infty}$ be generated from the fixed point iteration $\theta_{n+1} = T(\theta_n)$. If there exists a continuous function $f : \Theta \to \mathbb{R}$ satisfying that*

(i) *the sequence $\{f(\theta_n)\}_{n=1}^{\infty}$ converges, and*
(ii) *for $\theta \in \Theta$, if $f(T(\theta)) = f(\theta)$, then $T(\theta) = \theta$,*

then every converging subsequence of $\{\theta_n\}_{n=1}^{\infty}$ converges to a fixed point of T.

Proof. Let $\{\theta_{s_n}\}_{n=1}^{\infty}$ be a converging subsequence of $\{\theta_n\}_{n=1}^{\infty}$, where $\theta_{s_n} \to \theta^*$. Also let f^* be the limit of $f(\theta_n)$. Then $f(\theta_{s_n}) \to f(\theta^*)$; therefore $f(\theta^*) = f^*$. Meanwhile, from the continuity of T and f, we have $T(\theta_{s_n}) \to T(\theta^*)$ and $f(\theta_{s_n+1}) = f(T(\theta_{s_n})) \to f(T(\theta^*))$, which implies that $f(T(\theta^*)) = f^*$. Condition (ii) of the lemma now implies that $\theta^* = T(\theta^*)$. ∎

Our objective function E' is just one such helper function f, and the orthogonal polar factor function plays the role of the mapping T in the above lemma. The following lemma establishes the fact that the orthogonal polar factor function is continuous.

LEMMA 5.6. *The orthogonal polar factor function $g : A \to Q$ defined on the set of matrices with full column rank is continuous. Here Q is the orthogonal polar factor of $A \in \mathbb{R}^{m \times n}$, $m \geq n$.*

Proof. First, function g is well defined, since the orthogonal polar factor of a full rank matrix exists and is unique [21]. When Q and Q' are the orthogonal polar factors of A and A', respectively, Sun and Chen [39] have shown that

$$\|Q - Q'\|_F \leq \frac{2}{\|A^+\|_2} \|A - A'\|_F,$$

where $^+$ means pseudoinverse. Hence if $A_1, A_2, \ldots$ converges to A^*, then $g(A_1)$, $g(A_2), \ldots$ converges to $g(A^*)$. ∎

Now we are ready to prove the following result.

THEOREM 5.7. *Every converging subsequence of $\{U_{(p)}^{(1)}, \ldots, U_{(p)}^{(N)}\}_{p=1}^{\infty}$ generated by Algorithm* 1 *converges to a stationary point of the objective function E' defined in* (5.6)*, provided the matrices $V^{(n)}$ in line* 3 *of the algorithm do not become rank-deficient throughout the iterations.*

Proof. For convenience, let U denote the side-by-side concatenation of the $U^{(n)}$'s; i.e., at iteration number p we write $U_{(p)} := [U_{(p)}^{(1)}, \ldots, U_{(p)}^{(N)}]$. For each iteration, $V_{(p)}^{(n)}\Sigma^{(idx)}$ is computed from $U_{(p)}$ and polar factorized, and $U_{(p)}^{(n)}$ is updated. Let

T be the composite of all of these iterations running n from 1 to N. That is, $U_{(p+1)} = T(U_{(p)})$. It is not hard to see that T is continuous by Lemma 5.6. The objective function E' taking parameter $U_{(p)}$ has been previously shown such that the sequence $\{E'(U_{(p)})\}_{p=1}^{\infty}$ is monotonically converging.

Hence by Lemma 5.5, in order to prove this theorem, it will suffice to show that $E'(T(U)) = E'(U)$ implies $T(U) = U$. Then every converging subsequence of $\{E'(U_{(p)})\}_{p=1}^{\infty}$ converges to a fixed point, which satisfies the first order condition (5.19); i.e., it is also a stationary point of E'.

If $E'(T(U)) = E'(U)$, then formula (5.21) indicates that the σ_i values have not changed after the iteration. In particular, for any n, the update of $U^{(n)}$ has not changed $\mathrm{tr}({U^{(n)}}^T V^{(n)}\Sigma)$. Since the orthogonal polar factor of $V^{(n)}\Sigma$ is unique when $V^{(n)}$ is not rank-deficient, this means that $U^{(n)}$ has not changed. This in turn means that U is a fixed point of the mapping T. ∎

The condition in the theorem is not a strong requirement in general. Of course, the columns $v_i^{(n)}$ of the matrix $V^{(n)}$, as computed from (5.15), will be linearly dependent if the n-rank of $\mathcal{A}$ is less than r. For practical applications, the tensor usually has full n-ranks for all n, so this does not hamper the applicability of the theorem.

Though the global convergence of $\{U_{(p)}\}$ is not determined, when localized, it is possible that this parameter sequence converges. The following lemma and theorems consider this situation.

LEMMA 5.8. *If a sequence $\{\theta_n\}_{n=1}^{\infty}$ is bounded, and all of its converging subsequences converge to θ^*, then $\theta_n \to \theta^*$.*

Proof (by contradiction). If $\{\theta_n\}_{n=1}^{\infty}$ does not converge to θ^*, then there is an $\epsilon > 0$ such that there exists a subsequence $S = \{\theta_{s_n}\}_{n=1}^{\infty}$, where $\|\theta_{s_n} - \theta^*\| \geq \epsilon$ for all n. Since S is bounded, it has a converging subsequence S'. Then S' as a subsequence of $\{\theta_n\}_{n=1}^{\infty}$ converges to a limit other than θ^*. ∎

THEOREM 5.9. *Let $U_* = [U_*^{(1)}, \ldots, U_*^{(N)}]$ be a local maximum of the objective function E' defined in* (5.6)*. If the sequence $\{U_{(p)} := [U_{(p)}^{(1)}, \ldots, U_{(p)}^{(N)}]\}_{p=1}^{\infty}$ generated by Algorithm* 1 *lies in a neighborhood of U_*, where U_* is the only stationary point in that neighborhood, and if the full rank requirement in Theorem* 5.7 *is satisfied, then the sequence $\{U_{(p)}\}_{p=1}^{\infty}$ converges to U_*.*

Proof. This immediately follows from Theorem 5.7 and Lemma 5.8. ∎

Note that since the starting elements of a sequence have no effect on its convergence behavior, the above theorem holds whenever the tailing subsequence, starting from a sufficiently large p, lies within the neighborhood.

A weaker, but simpler, result is the following corollary.

COROLLARY 5.10. *Let $U_* = [U_*^{(1)}, \ldots, U_*^{(N)}]$ be a local maximum of the objective function E' defined in* (5.6)*. If this local maximum is unique and if the full rank requirement in Theorem* 5.7 *is satisfied, then the sequence $\{U_{(p)} := [U_{(p)}^{(1)}, \ldots, U_{(p)}^{(N)}]\}_{p=1}^{\infty}$ generated by Algorithm* 1 *converges to U_*.*

5.6. LROAT for symmetric tensors. An order-N tensor $\mathcal{A} \in \mathbb{R}^{d \times d \times \cdots \times d}$, whose dimensions of all modes are the same, is *symmetric* if, for all permutations π,

$$a_{i_1, i_2, \ldots, i_N} = a_{i_{\pi(1)}, i_{\pi(2)}, \ldots, i_{\pi(N)}}.$$

For symmetric tensors, usually the approximation problem (5.4) has an additional constraint that the side-matrices $U^{(n)}$'s are the same for all n; i.e., the problem

becomes

$$\begin{aligned} \min \quad & E = \left\| \mathcal{A} - \sum_{i=1}^{r} \sigma_i u_i \otimes u_i \otimes \cdots \otimes u_i \right\|_F \\ \text{s.t.} \quad & \langle u_j, u_k \rangle = \delta_{jk}. \end{aligned} \tag{5.22}$$

Applying similar arguments to those in section 5.1, it is easily seen that (5.22) is equivalent to the following problem:

$$\begin{aligned} \max \quad & E' = \sum_{i=1}^{r} \left(\mathcal{A} \times_1 u_i^T \times_2 u_i^T \times \cdots \times_N u_i^T \right)^2 \\ \text{s.t.} \quad & \langle u_j, u_k \rangle = \delta_{jk}. \end{aligned} \tag{5.23}$$

The supremum of E' can be attained. Further, the "maximal-diagonality" decomposition of $\mathcal{A}$ (cf. (5.12)) has an additional property that the core $\mathcal{S}$ is symmetric. Also, the first order condition (5.19) is simplified to

$$V\Sigma = UM.$$

Hence, there are two approaches to compute the approximation for the symmetric tensor $\mathcal{A}$. The first approach is to directly apply LROAT on $\mathcal{A}$. Theorems in the above section guarantee the convergence under mild assumptions, but the side-matrices might no longer be the same, though in the next section an experiment indicates that they indeed converge to the same matrix. The second approach is to only use a single initial guess U and omit the for-loop on n (line 2 of Algorithm 1). We call this the *symmetric variant of LROAT*. In this case Theorem 5.4 no longer holds, i.e., the iterations might not monotonically increase the objective value E' defined in (5.23), since the for-loop on n is omitted. An experiment in the next section shows an oscillating phenomenon, which is similar to the one indicated in Figure 4.1 of [27], for the objective value E'.

6. Numerical experiments. This section will show a few experiments to illustrate the convergence behavior and the approximation quality of LROAT. For comparisons see the ALS methods for Tucker and PARAFAC, whose implementations are based on the codes from the MATLAB Tensor Toolbox developed by Bader and Kolda [2]. We use the major left singular vectors of the unfoldings as the initial guess input for all of the algorithms compared. When it comes to the quality of the final approximation, experience shows that compared with random orthonormal vectors, singular vectors as initial guesses do not offer any advantage. It has been argued that running the algorithms several times using different sets of random initial guess enhances the probability of hitting the global optimum. We use singular vectors here only for repeatability.

6.1. Convergence of LROAT. In the first experiment, we test LROAT (and the symmetric variant of LROAT mentioned in section 5.6) on a few tensors listed in Table 6.1. The results are shown in Figures 6.1 and 6.2. Each row of the figures is one test on a tensor. The left plot shows the objective value E' (the same as the norm of the approximated tensor $\mathcal{T}$) for each iteration p, while the right plot shows the convergence history of the $U^{(n)}$'s. Since the optima are unknown, we plot the values $\|U_{(p)}^{(n)} - U_{(p-1)}^{(n)}\|_F$ to indicate the convergence of the sequence $\{U_{(p)}^{(n)}\}_{p=1,2,\ldots}$. Since

these values are plotted on logarithmic scales, if the curves are bounded from above by a straight decreasing line, then it is indicated that the convergence of the sequence is at least linear.

TABLE 6.1
The tensors used for the first experiment. The value r is the rank input to LROAT; it is not the rank of the tensor.

Tensor	Dimensions	r	Notes
$\mathcal{A}_1$	$20 \times 16 \times 10 \times 32$	5	random tensor
$\mathcal{A}_2$	$20 \times 16 \times 10 \times 32$	5	rank-5 tensor + Gaussian noise
$\mathcal{A}_3$	$10 \times 10 \times 10$	5	the (i,j,k)-entry $= 1/(i^2+j^2+k^2)$
$\mathcal{A}_4$	$3 \times 3 \times 3 \times 3$	2	see [27, Example 1]

Figures 6.1 and 6.2 show a total of five tests. The first test (Figure 6.1(a)) uses a randomly generated tensor $\mathcal{A}_1$. The second test (Figure 6.1(b)) uses a low-rank-plus-Gaussian-noise tensor

$$\mathcal{A}_2 = \mathcal{B}_1 + \rho\mathcal{B}_2,$$

where the low rank tensor $\mathcal{B}_1$ is in the form (2.3) with $r = 5$, the Gaussian noise tensor $\mathcal{B}_2$ has normally distributed elements, and $\rho = 0.1 \|\mathcal{B}_1\|_F / \|\mathcal{B}_2\|_F$. In these two tests the two tensors are applied to the LROAT algorithm. The third test (Figure 6.1(c)) uses a symmetric tensor $\mathcal{A}_3$ with entries

$$(\mathcal{A}_3)_{ijk} = \frac{1}{i^2+j^2+k^2}.$$

In this test $\mathcal{A}_3$ is applied to the symmetric variant of LROAT. All three tests show a linear convergence rate. The fourth (Figure 6.2(a)) and the fifth (Figure 6.2(b)) tests use a symmetric tensor $\mathcal{A}_4$ introduced in [27, Example 1]:

$$\begin{aligned}
&(\mathcal{A}_4)_{1111} = 0.2883, && (\mathcal{A}_4)_{1112} = -0.0031, && (\mathcal{A}_4)_{1113} = 0.1973,\\
&(\mathcal{A}_4)_{1112} = -0.2485, && (\mathcal{A}_4)_{1123} = -0.2939, && (\mathcal{A}_4)_{1133} = 0.3847,\\
&(\mathcal{A}_4)_{1222} = 0.2972, && (\mathcal{A}_4)_{1223} = 0.1862, && (\mathcal{A}_4)_{1233} = 0.0919,\\
&(\mathcal{A}_4)_{1333} = -0.3619, && (\mathcal{A}_4)_{2222} = 0.1241, && (\mathcal{A}_4)_{2223} = -0.3420,\\
&(\mathcal{A}_4)_{2233} = 0.2127, && (\mathcal{A}_4)_{2333} = 0.2727, && (\mathcal{A}_4)_{3333} = -0.3054.
\end{aligned}$$

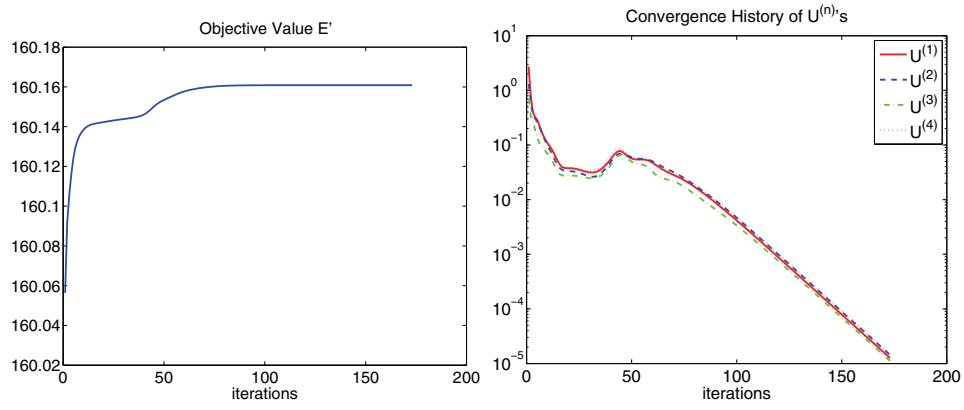
In [27], the symmetric higher order power method for computing the optimal rank-1 approximation of $\mathcal{A}_4$ is shown to be nonconverging. We experiment with this tensor with $r = 2$ on LROAT and the symmetric variant of LROAT. Figure 6.2(a) shows that when applied to LROAT, the approximation to $\mathcal{A}_4$ indeed linearly converges, and what is more, all of the side-matrices converge to the same result. The approximation computed by LROAT is

$$\mathcal{A}_4 \approx \sigma_1 u^{(1)} \otimes u^{(1)} \otimes u^{(1)} \otimes u^{(1)} + \sigma_2 u^{(2)} \otimes u^{(2)} \otimes u^{(2)} \otimes u^{(2)}$$

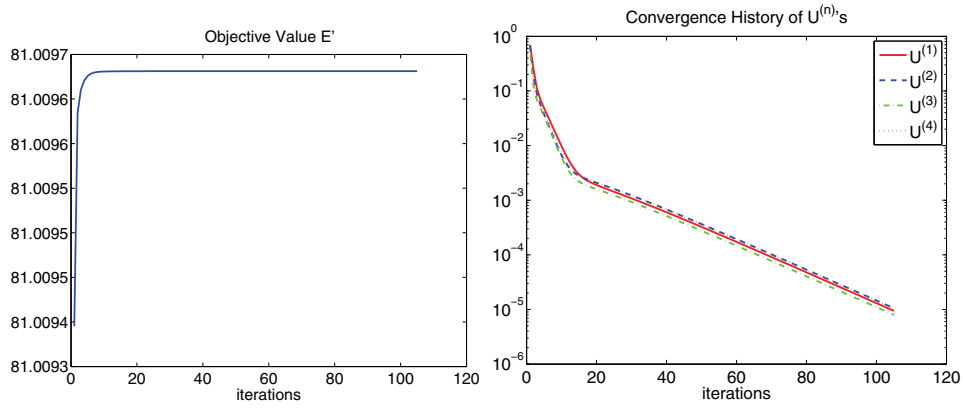
with

$$\begin{aligned}
&\sigma_1 = -1.0939, && u^{(1)} = \begin{bmatrix} -0.5946 & 0.7503 & 0.2890 \end{bmatrix}^T,\\
&\sigma_2 = -0.55594, && u^{(2)} = \begin{bmatrix} 0.1947 & -0.2144 & 0.9572 \end{bmatrix}^T.
\end{aligned}$$

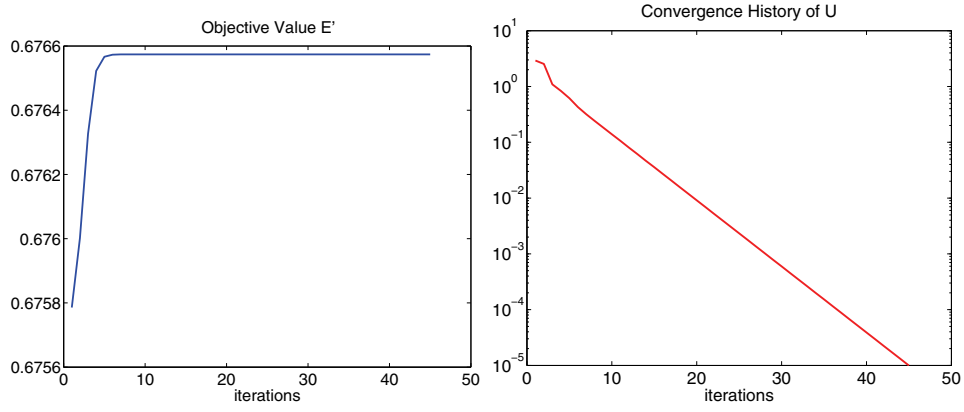
On the other hand, Figure 6.2(b) shows that the symmetric variant of LROAT fails to converge.



(a) Tensor $\mathcal{A}_1$: Randomly generated.

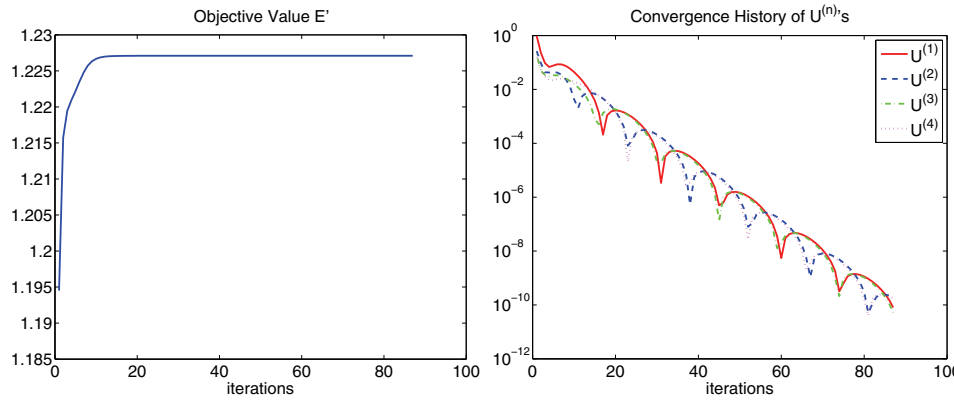


(b) Tensor $\mathcal{A}_2$: Low rank plus Gaussian noise.

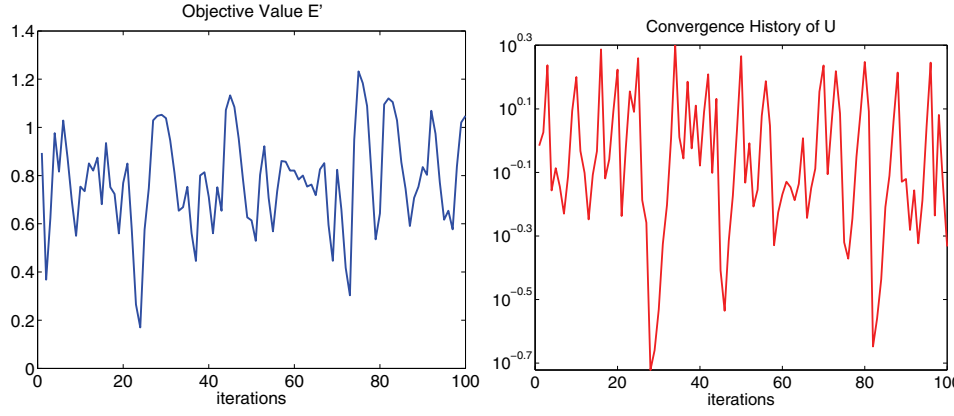


(c) Tensor $\mathcal{A}_3$: $(\mathcal{A}_3)_{ijk} = 1/(i^2 + j^2 + k^2)$.

FIG. 6.1. *Experiment* 1*: Convergence tests for LROAT.*



(a) Tensor $\mathcal{A}_4$ directly applied to LROAT.



(b) Tensor $\mathcal{A}_4$ applied to the symmetric variant of LROAT.

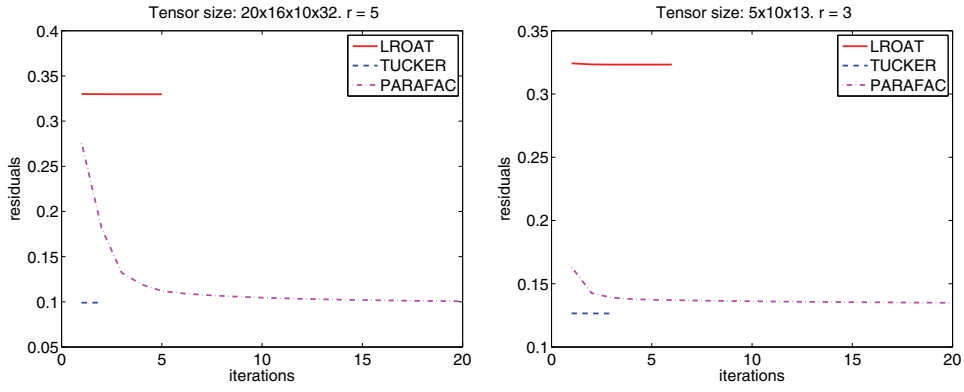
FIG. 6.2. *Experiment* 1 *(continued): Convergence tests for LROAT.*

6.2. Low rank orthogonal approximation compared with Tucker and PARAFAC. In the second experiment, we compare the approximation quality of three different models: low rank orthogonal approximation (without confusion in this section, we call this model "LROAT," which happens to be the name of the algorithm, for short), Tucker, and PARAFAC. See Figure 6.3. We experiment with two tensors: a low-rank-plus-Gaussian-noise tensor which is generated the same way as $\mathcal{A}_2$ and a real-life tensor. The latter is obtained from a problem in acoustics [20], and the data can be downloaded from [16]. The residual norms

$$res(p) := \frac{\left\|\mathcal{A} - \mathcal{T}_{(p)}\right\|_F}{\|\mathcal{A}\|_F}$$

over all of the iterations p are plotted.

Figure 6.3 indicates three facts: (1) the three models approximate the data tensor well to some extent (less than 35% of the information is lost due to approximation); (2) PARAFAC is usually slow to converge; (3) the residual norm for LROAT is larger than those of Tucker and PARAFAC. The last fact is not unexpected since LROAT can be considered a special case of Tucker and of PARAFAC: The Tucker model has a



(a) Low-rank-plus-Gaussian-noise tensor. (b) Real-life tensor.

FIG. 6.3. *Experiment 2: Comparison of LROAT, Tucker, and PARAFAC.*

full core while the core for LROAT is diagonal, and unlike LROAT the side-matrices in the PARAFAC model are not restricted to be orthogonal.

6.3. An application. In the blind source separation (BSS) problem [5], the cumulant tensor of order 4 is a rank-R tensor:

$$\sum_{i=1}^{R} \sigma_i u_i \otimes u_i \otimes u_i \otimes u_i, \tag{6.1}$$

where R is the number of sources and u_i is the ith column of the mixing matrix. In the prewhitening approach for the BSS problem, the u_i's become the columns of the composite of the whitening matrix and the mixing matrix; that is, the u_i's are length-R vectors and are orthonormal. Hence, this prewhitening approach reduces to computing the tensor SVD of the cumulant tensor. Since in practice this tensor is estimated from a finite data set, it is not exact. Thus, the low rank orthogonal approximation becomes a suitable tool to recover the u_i's.

In an experiment, we let $R = 3$ and generate a data tensor

$$\mathcal{A}_5 = \mathcal{B}_3 + \rho \mathcal{B}_4,$$

where $\mathcal{B}_3$ is as (6.1), $\mathcal{B}_4$ is a symmetric tensor with normally distributed elements, and $\rho = 0.05 \|\mathcal{B}_3\|_F / \|\mathcal{B}_4\|_F$. The σ_i's are

$$\sigma_1 = 0.7942, \quad \sigma_2 = 0.5678, \quad \sigma_3 = 0.4611,$$

and the u_i's are

$$U = [u_1, u_2, u_3] = \begin{bmatrix} 0.0974 & 0.4049 & 0.9092 \\ 0.9918 & -0.1154 & -0.0548 \\ 0.0827 & 0.9071 & -0.4128 \end{bmatrix}.$$

We use four methods to compute the rank-R (or rank-(R, R, R)) approximations to $\mathcal{A}_5$: LROAT, incremental rank-1 approximation, PARAFAC, and Tucker. All four

methods return same side-matrices for all modes. They are

$$U_{\text{LROAT}} = \begin{bmatrix} 0.0937 & 0.3822 & 0.9193 \\ 0.9918 & -0.1164 & -0.0527 \\ 0.0869 & 0.9167 & -0.3899 \end{bmatrix}, \quad U_{\text{inc}} = \begin{bmatrix} 0.0841 & 0.3795 & 0.9162 \\ 0.9929 & -0.1282 & -0.0745 \\ 0.0846 & 0.9163 & -0.3938 \end{bmatrix},$$

$$U_{\text{PARAFAC}} = \begin{bmatrix} 0.0841 & 0.3795 & 0.9162 \\ 0.9929 & -0.1282 & -0.0745 \\ 0.0846 & 0.9163 & -0.3938 \end{bmatrix},$$

$$U_{\text{Tucker}} = \begin{bmatrix} 0.0627 & 0.3707 & 0.9266 \\ 0.9952 & -0.0937 & -0.0298 \\ 0.0758 & 0.9240 & -0.3748 \end{bmatrix}.$$

Observations are as follows:

1. The U_{inc} and U_{PARAFAC} are not orthogonal.
2. Compared with Tucker, LROAT gives better approximations to the vectors u_i's:

$$\|U - U_{\text{LROAT}}\| = 0.0252, \quad \|U - U_{\text{Tucker}}\| = 0.0527.$$

3. In terms of approximation quality, the residual norms (in percentage of the norm of $\mathcal{A}_5$), are

$$res_{\text{LROAT}} = 3.07\%, \quad res_{\text{inc}} = 1.36\%, \quad res_{\text{PARAFAC}} = 1.36\%, \quad res_{\text{Tucker}} = 0\%.$$

7. Concluding remarks. In the present paper we studied the tensor SVD and characterized its existence in relation to the HOSVD. Similar to the concept of rank, the SVD of higher order tensors exhibits a quite different behavior and characteristics from those of matrices. Thus, the SVD of a matrix is guaranteed to exist, though it may have different representations due to orthogonal transformations of singular vectors corresponding to the same singular value. On the other hand, there are many ways in which a tensor can fail to have an SVD (see the results in section 4), but when it exists, this decomposition is unique up to signs.

We have also discussed a new form of optimal low rank approximation of tensors, where orthogonality is required. This approximation is inspired by the constraints of the Tucker model and the PARAFAC model. In some applications, the proposed approximation model may be favored, since it results in N sets of orthonormal vectors or, equivalently, r mutually orthogonal unit rank-1 tensors with different weights. Among the advantages of this approximation over the Tucker model is the fact that it requires far fewer entries to represent the core, and that it is easier to interpret. Also, compared with the PARAFAC model, the orthogonality of vectors may be useful in some cases. Further, the LROAT algorithm for computing the proposed approximation does not seem to exhibit the well-known slow convergence from which the ALS algorithm for PARAFAC suffers.

A major restriction of the proposed model is that the number of terms r can not exceed the smallest dimension of all modes of the tensor. A consequence is that the approximation may still be very different from the original tensor even when the maximum r is employed. However, we note that when performing data analysis, the

interpretation of the vectors and the core tensor might be more important than how much is lost when the data is approximated.

A nice aspect of the proposed approximation is that the optimum of the objective function can theoretically be attained, in contrast to the PARAFAC model which is ill-posed in a strict mathematical sense. We presented an algorithm to compute this approximation, but the computed result is optimal only in a local neighborhood. It will be interesting to study for what tensors or what initial guesses the LROAT algorithm converges to the global optimum, or to devise a new algorithm to solve this optimization problem. It is an open problem how fast LROAT converges, although empirically convergence is observed to be linear. We also discussed the symmetric variant of LROAT and pointed out the possibility of its nonconvergence. Hence the convergence properties of this variant, and the observed phenomenon that the original LROAT algorithm can yield same side-matrices for symmetric tensors, remain to be investigated.

Appendix. Does the ALS algorithm for PARAFAC converge? It has been pointed out that the ALS algorithm for computing the PARAFAC model may converge very slowly due to degenerate solutions or multicollinearities, and many alternatives have been proposed to address this problem [36, 37, 26]. During iterations, the objective value monotonically decreases by the nature of the ALS procedure, and since the sequence is bounded, it converges. However, a proof of the convergence of the parallel factors is lacking. In general it is assumed that these factors converge, but may take a very large number of iterations. In this section, we discuss an experiment showing that the general concept of convergence is unclear in this context. Though only one example is given, we note that the exhibited behavior is not rare for randomly generated tensors. (On the other hand it may be argued that tensors in real applications are far from being filled with random entries.)

We generate an order-3 tensor $\mathcal{A} \in \mathbb{R}^{3\times3\times3}$ and run the ALS algorithm on $r = 2$, i.e., to compute the approximation

$$\mathcal{A} \approx \lambda_1 u_1^{(1)} \otimes u_1^{(2)} \otimes u_1^{(3)} + \lambda_2 u_2^{(1)} \otimes u_2^{(2)} \otimes u_2^{(3)}.$$

The MATLAB code which generates the tensor $\mathcal{A}$ is as follows:

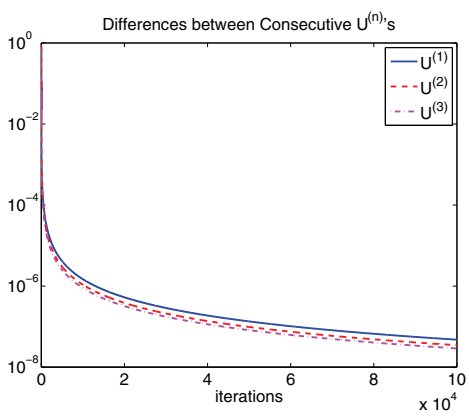
```
A(:,:,1) = [.99 .29 .08; .44 .69 .19; .00 .49 .97];
A(:,:,2) = [.36 .64 .10; .13 .73 .89; .01 .02 .76];
A(:,:,3) = [.58 .55 .98; .68 .77 .04; .96 .61 .98];
```

We use $u_i^{(n)} = e_i$, where $n = 1, 2, 3$ and e_i is the ith column of the identity matrix, as the initial guess.

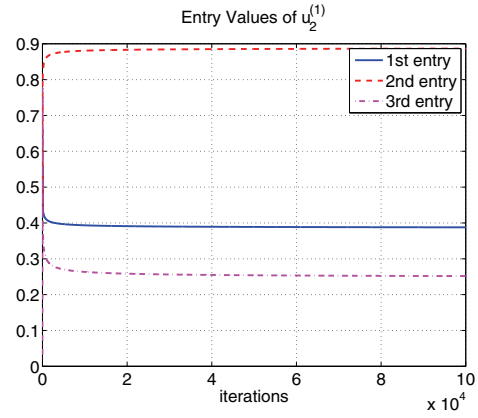
Denote $U^{(n)} = [u_1^{(n)}, u_2^{(n)}]$ for $n = 1, 2, 3$. Two plots are shown after running 10^5 iterations (see Figure A.1). Figure A.1(a) shows the "convergence" history for each $U^{(n)}$. The curves represent $\|U_{(p)}^{(n)} - U_{(p-1)}^{(n)}\|$, where p is the index of the iterations. A necessary condition for convergence to occur is that all three curves decrease to zero. However, we see from the figure that this may not be the case. To test the conjecture that each of the curves tends to a nonzero value, we use the expression

$$\log_{10} y = \frac{a}{(10^{-4}x)^{1/\alpha}} + b$$

to fit the tailing part of the curves (starting from the 2×10^4th iteration). Table A.1 gives the fitting results for different α's. When the number of iterations tends to



(a) Differences of the $U^{(n)}$'s between consecutive iterations.



(b) Entry values of the parallel factor $u_2^{(1)}$.

FIG. A.1. *Slow convergence or nonconvergence of ALS for PARAFAC.*

TABLE A.1

Curve fitting for the three curves in Figure A.1(a) *using different* α *values. The error is measured as the quadratic mean of fitting errors in logarithmic scales, i.e., the RMS of* $|\log_{10} y - \log_{10} y_{fit}|$. *It will be easier to understand this error by noticing that the vertical axis of Figure* A.1(a) *has a length* 8 *(after taking logarithm).*

	α	1	2	3	4	5
$U^{(1)}$	a	2.8407	2.7925	3.2657	3.8359	4.4404
	b	−7.5060	−8.1549	−8.8047	−9.4544	−10.1042
	error	0.0595	0.0301	0.0201	0.0151	0.0121
$U^{(2)}$	a	2.8408	2.7926	3.2658	3.8360	4.4406
	b	−7.6428	−8.2917	−8.9415	−9.5913	−10.2411
	error	0.0595	0.0301	0.0201	0.0151	0.0121
$U^{(3)}$	a	2.8411	2.7930	3.2662	3.8365	4.4411
	b	−7.7219	−8.3709	−9.0208	−9.6707	−10.3205
	error	0.0595	0.0301	0.0201	0.0151	0.0121

infinity, the value 10^b will show the limit of the differences between two consecutive $U^{(n)}$'s.

It is still difficult to conclude for this example that the iterations do not converge since rounding has not been taken into account. However, it makes no practical difference for this case whether the sequence actually converges or whether it is exceedingly slow to converge. The result, if convergence holds, will be an inordinate number of iterations to reach a desirable level of convergence, and the cost will be too high in practice. This can be made evident by examining Figure A.1(b), which plots the parallel factor $u_2^{(1)}$ over all iterations: The 3rd entry of $u_2^{(1)}$ decreases from 0.2540 at the 5×10^4th iteration to 0.2517 at the 10^5th iteration.

REFERENCES

[1] E. ACAR, S. A. CAMTEPE, M. KRISHNAMOORTHY, AND B. YENER, *Modeling and multiway analysis of chatroom tensors*, in Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI 05), 2005.

[2] B. W. BADER AND T. G. KOLDA, *Algorithm* 862*: MATLAB tensor classes for fast algorithm prototyping*, ACM Trans. Math. Software, 32 (2006), pp. 635–653.

[3] J. BERGE, J. DE LEEUW, AND P. M. KROONENBERG, *Some additional results on principal*

components analysis of three-mode data by means of alternating least squares algorithms, Psychometrika, 52 (1987), pp. 183–191.

[4] R. Bro, *Review on multiway analysis in chemistry*—2000–2005, Crit. Rev. Anal. Chem., 36 (2006), pp. 279–293.

[5] J.-F. Cardoso and P. Comon, *Independent component analysis, a survey of some algebraic methods*, in Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS 96), 1996.

[6] J. D. Carroll and J.-J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[7] P. Comon, *Independent component analysis, a new concept?*, Signal Process., 36 (1994), pp. 287–314.

[8] P. Comon, *Tensor decompositions*, in Mathematics in Signal Processing V, J. G. McWhirter and I. K. Proudler, eds., Oxford University Press, Oxford, UK, 2002, pp. 1–24.

[9] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, *Symmetric tensors and symmetric tensor rank*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1254–1279.

[10] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*, Ph.D. thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 1997.

[11] L. De Lathauwer, *A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666.

[12] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[13] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-*1 *and rank-*$(R_1, R_2, \ldots, R_N)$ *approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[14] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Independent component analysis and (simultaneous) third-order tensor diagonalization*, IEEE Trans. Signal Process., 49 (2001), pp. 2262–2271.

[15] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 295–327.

[16] B. De Moor, *DaISy: Database for the Identification of Systems*, http://homes.esat.kuleuven.be/∼smc/daisy/, Used dataset: `tongue.dat`, section: Biomedical Systems, code: 97-001.

[17] V. de Silva and L.-H. Lim, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1084–1127.

[18] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[19] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[20] R. A. Harshman, P. Ladefoged, and L. Goldstein, *Factor analysis of tongue shapes*, J. Acoust. Soc. Amer., 62 (1977), pp. 693–707.

[21] N. J. Higham and R. S. Schreiber, *Fast polar decomposition of an arbitrary matrix*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 648–655.

[22] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[23] J. JáJá and J. Takche, *On the validity of the direct sum conjecture*, SIAM J. Comput., 15 (1986), pp. 1004–1020.

[24] H. A. L. Kiers, *TUCKALS core rotations and constrained TUCKALS modelling*, Statistica Applicata, 4 (1992), pp. 659–667.

[25] H. A. L. Kiers, *An alternating least squares algorithms for PARAFAC*2 *and three-way DEDICOM*, Comput. Statist. Data Anal., 16 (1993), pp. 103–118.

[26] H. A. L. Kiers, *A three-step algorithm for CANDECOMP/PARAFAC analysis of large data sets with multicollinearity*, J. Chemometrics, 12 (1998), pp. 155–171.

[27] E. Kofidis and P. A. Regalia, *On the best rank-*1 *approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 863–884.

[28] T. G. Kolda, *Orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 243–255.

[29] T. G. Kolda, *A counterexample to the possibility of an extension of the Eckart–Young low-rank approximation theorem for the orthogonal rank tensor decomposition*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 762–767.

[30] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., to appear.
[31] T. G. Kolda, B. W. Bader, and J. P. Kenny, *Higher-order web link analysis using multilinear algebra*, in Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 05), 2005.
[32] W. P. Krijnen, *Convergence of the sequence of parameters generated by alternating least squares algorithms*, Comput. Statist. Data Anal., 51 (2006), pp. 481–489.
[33] P. M. Kroonenberg and J. De Leeuw, *Principal component analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 45 (1980), pp. 69–97.
[34] S. E. Leurgans, R. T. Ross, and R. B. Abel, *A decomposition for three-way arrays*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1064–1083.
[35] C. D. Moravitz Martin and C. F. Van Loan, *A Jacobi-type method for computing orthogonal tensor decompositions*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1219–1232.
[36] B. C. Mitchell and D. S. Burdick, *Slowly converging PARAFAC sequences: Swamps and two-factor degeneracies*, J. Chemometrics, 8 (1994), pp. 155–168.
[37] P. Paatero, *A weighted non-negative least squares algorithm for three-way "PARAFAC" factor analysis*, Chemometrics Intell. Lab. Syst., 38 (1997), pp. 223–242.
[38] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis: Applications in the Chemical Sciences*, Wiley, Chichester, 2004.
[39] J. Sun and C. Chen, *Generalized polar decomposition*, Math. Numer. Sinica, 11 (1989), pp. 262–273.
[40] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.
[41] M. A. O. Vasilescu and D. Terzopoulos, *Multilinear subspace analysis for image ensembles*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 03), 2003.
[42] H. Wang, Q. Wu, L. Shi, Y. Yu, and N. Ahuja, *Out-of-core tensor approximation of multi-dimensional matrices of visual data*, ACM Trans. Graphics, 24 (2005), pp. 527–535.
[43] T. Zhang and G. H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1735–1760

STATE FEEDBACK DECOUPLING PROBLEM WITH STABILITY FOR (A, B, C, D) QUADRUPLES*

DELIN CHU†, M. MALABRE‡, AND ROGER C. E. TAN†

Abstract. The state feedback decoupling problem with stability for general proper systems described by (A, B, C, D) quadruples has been studied for a long time. But, it is still an open problem in the sense that there is still a lack of numerically verifiable solvability conditions and numerically implementable methods for solving it in the existing literature. In this paper numerically verifiable solvability conditions and a numerical method for solving this open problem are developed. The proposed method is based on orthogonal transformations and hence can be implemented in a numerically reliable manner.

Key words. decoupling, stability, (A, B, C, D) quadruple

AMS subject classifications. Primary, 65; Secondary, 15A18, 15A24

DOI. 10.1137/070692716

1. Introduction.

Decoupling is a fundamental objective in control theory [6, 11, 12, 16]. It is easy for an operator to control a system if each input of the system can affect only one of the outputs. Usually, a given multivariable system has couplings. A strategy is then to design a controller such that the resulting control system has no couplings between the inputs and the outputs. Decoupling control is popular not only because it can simplify multivariable system design and control but also because it is a desired feature in many practical applications, at least in process and chemical industry [6, 11, 12]. Although the exact decoupling is not always possible to achieve, it has been pointed out in [23, p. 796] that the state feedback decoupling (namely, the diagonal decoupling by state feedback in [23]) is likely to be feasible for a wide variety of general proper systems of the form (1.1) below. Moreover, the decoupling technique has found applications not only in process and chemical industry [6, 11, 12, 20] but also in industrial robots [16, 17, 28, 33, 36], and it has been studied extensively in the last three decades; see [6, 8, 11, 12, 15, 16, 18, 23, 24, 25, 26, 28, 29, 35, 39, 41, 44, 45, 46, 47, 48, 49, 50] and the references therein. Hence, decoupling is of importance in systems design and control.

Consider a proper multivariable system described by an (A, B, C, D) quadruple:

$$\begin{cases} \dot{x} = Ax + Bu, \\ y = Cx + Du, \end{cases} \tag{1.1}$$

where $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, $C \in \mathbf{R}^{m\times n}$, $D \in \mathbf{R}^{m\times m}$, D is the direct feedthrough matrix, $x \in \mathbf{R}^n$ is the state, $u \in \mathbf{R}^m$ is the control input, and $y \in \mathbf{R}^m$ is the output. If we apply the state feedback of the form

$$u = Fx + Hv \tag{1.2}$$

*Received by the editors May 23, 2007; accepted for publication (in revised form) by P. Benner July 3, 2008; published electronically February 20, 2009.
http://www.siam.org/journals/simax/30-4/69271.html

†Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543 (matchudl@math.nus.edu.sg, mattance@math.nus.edu.sg). The research of these authors was supported by NUS Research grant R-146-000-016-112.

‡IRCCyN (Institut de Recherche en Communications et Cybernetique de Nantes), CNRS UMR 6597, B.P. 92101, F-44321 Nantes, Cedex 3, France (Michel.Malabre@irccyn.ec-nantes.fr).

to (1.1), then the closed-loop system becomes

$$\begin{cases} \dot{x} = (A + BF)x + BHv, \\ y = (C + DF)x + DHv. \end{cases} \tag{1.3}$$

The transfer matrix from output y to input v in (1.3) is $(C+DF)(sI-A-BF)^{-1}BH+DH$. The state feedback decoupling problem (i.e., row by row decoupling problem) with stability studied in this paper can be formulated mathematically as follows.

DEFINITION 1.1. *The state feedback decoupling problem with stability for proper system* (1.1) *is solvable if there exist matrices* $F \in \mathbf{R}^{m\times n}$ *and* $H \in \mathbf{R}^{m\times m}$ *such that*

$$(C + DF)(sI - A - BF)^{-1}BH + DH \text{ is nonsingular and diagonal}, \tag{1.4}$$

and $A+BF$ *is stable (i.e., all eigenvalues of* $A+BF$ *are on the open left half complex plane* $\mathbf{C}^-$*).*

Up to now, many important contributions have been made to the study of the state feedback decoupling problem with stability for the strictly proper system

$$\begin{cases} \dot{x} = Ax + Bu, \\ y = Cx, \end{cases} \tag{1.5}$$

which has no direct feedthrough matrix; in particular, we have the following:

• Solvability conditions have been established in [49, 50]. In [49, 50], the class of all feedback matrices which decouple the system (1.5) and the number of closed-loop poles which can be assigned are characterized. But, as pointed out in [4, p. 34], the conditions given in [49, 50] are difficult to apply.

• Solvability conditions have also been given in [24, 25, 26, 47] based on the geometric/structural approaches.

• Solvability conditions have also been given in [21] based on a polynomial equations approach; these conditions are implicit and valid only for system (1.5) satisfying that the maximal (A, B)-controllability subspace lying in $\mathrm{Ker}(C)$ is the same as the maximal (A, B)-invariant subspace contained in $\mathrm{Ker}(C)$ [39], which implies that there is no extra difficulty in requiring internal stability of the decoupled closed-loop system.

• The family of all attainable transfer function matrices for the decoupled closed-loop system has been characterized in [4] under the assumption that the solvability conditions in [25] are satisfied; that is, system (1.5) is decouplable by a state feedback of the form (1.2). Based on the results in [4], a numerical algorithm has been developed in [7] for computing a desired feedback matrix pair (F, H). The main steps of this algorithm include the following:

(1) Compute the global and row infinite zeros, and global and row invariant finite zeros of system (1.5);

(2) compute the last invariant polynomial $z_i(s)$ of the matrix $\left[\begin{smallmatrix} sI-A & B \\ c_i & 0 \end{smallmatrix}\right]$ with c_i being the ith row of C $(i = 1, \ldots, m)$, and then define

$$W_1(s) = \begin{bmatrix} k_1 \frac{z_1(s)}{a_1(s)} & & \\ & \ddots & \\ & & k_m \frac{z_m(s)}{a_m(s)} \end{bmatrix},$$

where $k_1, \ldots, k_m$ are real numbers, for $i = 1, \ldots, m$, $a_i(s)$ is a monic polynomial with arbitrary roots and satisfying

$$\deg(a_i(s)) - \deg(z_i(s)) = n_i,$$

and $n_1, \ldots, n_m$ are the row infinite zero orders of the system (1.5);

(3) let $Q(s) = \left(C(sI - A)^{-1}B\right)^{-1} W_1(s)$, compute $G = \lim_{s\to\infty} Q(s)$, and then compute a basis for the left constant kernel of the matrix

$$\begin{bmatrix} (sI - A)^{-1}B \\ I - GQ^{-1}(s) \end{bmatrix}$$

via computing the left constant kernel of the matrix

$$\begin{bmatrix} \det(Q(s))\mathrm{adj}(sI - A)B \\ \det(sI - A)(\det(Q)I - G\mathrm{adj}(Q)) \end{bmatrix}.$$

But, (i) infinite zeros of system (1.5) are very sensitive to the perturbations including rounding errors to matrices A, B, and C; thus, the computation of the global and row infinite zeros of system (1.5) must be avoided if possible; and (ii) the computations of the last invariant polynomials $z_i(s)$ $(i = 1, \ldots, m)$, basis for the left constant kernel of the matrix

$$\left[(sI - A)^{-1}BI - GQ^{-1}(s)\right],$$

and the matrix

$$[\det(Q(s))\mathrm{adj}(sI - A)B\det(sI - A)(\det(Q)I - G\mathrm{adj}(Q))]$$

and its left constant kernel are very difficult and very expensive, and thus, the computational complexity of the algorithm in [7] is too high.

• A numerically reliable method, which does not compute the finite/infinite structures of system (1.5) explicitly, has been developed in [9] based on orthogonal transformations for solving this decoupling problem.

However, to the best of our knowledge, the results in [4, 7, 9, 21, 24, 25, 26, 47, 49, 50] cannot be generalized easily to the state feedback decoupling problem with stability for general proper systems of the form (1.1). In addition, the state feedback decoupling problem *without stability* for descriptor systems of the form

$$\begin{cases} E\dot{x} = Ax + Bu, \\ y = Cx, \end{cases} \tag{1.6}$$

with E *singular* has also been investigated in [2]. But, the state feedback decoupling problem *with stability* is not considered and still remains an open problem in [2].

The solvability condition for the state feedback decoupling problem *without stability* for general proper systems of the form (1.1) has been available in [23, 37], which is included in the following lemma.

LEMMA 1.2 (cf. [23, 37]). *Given $A \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{n\times m}, C \in \mathbf{R}^{m\times n}, D \in \mathbf{R}^{m\times m}$, let the matrix $L \in \mathbf{R}^{m\times m}$ be defined as follows: If the ith row of the matrix D is nonzero, this becomes the ith row of the matrix L, and set $l_i = 0$; otherwise, find the lowest positive integer l_i, for which the ith row of $CA^{l_i-1}B$ is nonzero; this then becomes the ith row of the matrix L. Then there exist matrices $F \in \mathbf{R}^{m\times n}$ and $H \in \mathbf{R}^{m\times m}$ such that* (1.4) *holds (i.e., $(C + DF)(sI - A - BF)^{-1}BH + DH$ is nonsingular and diagonal) if and only if the matrix L is nonsingular.*

The state feedback decoupling problem with stability for general proper systems of the form (1.1) is much more complicated than the state feedback decoupling problem without stability and has been considered in [23, 37, 41, 44, 46]. In [41, 46] only

sufficient conditions are given, while the necessary and sufficient conditions in [44] are based on the existence of some particular but unknown invariant subspaces of system (1.1) and some particular but also unknown solutions of the state feedback decoupling problem *without stability* for system (1.1); hence, the conditions given in [44] are not numerically verifiable. Furthermore, the results in [41, 44, 46] cannot give any direction to establish a numerically implementable method for solving the underlying problem in the general setting. In [23, 37], an algorithm for the state feedback decoupling problem for system (1.1) is presented. This algorithm can be summarized as follows:

• First, define the matrix L and the integers $\{l_i\}$ as those in Lemma 1.2, and let

$$\hat{T}(s) = \begin{bmatrix} p_1(s) & & \\ & \ddots & \\ & & p_m(s) \end{bmatrix},$$

where $p_i(s)$ $(i = 1, \ldots, m)$ are properly selected stable monic polynomials of degree l_i, i.e.,

$$p_i(s) = s^{l_i} + \text{lower} - \text{degree terms},$$

such that $\lim_{s\to\infty} \hat{T}(s)[C(sI - A)^{-1}B + D] = L$ and that $\{A, B, \hat{C}, L\}$ is a realization of $\hat{T}(s)[C(sI - A)^{-1}B + D]$ for some matrix $\hat{C}$.

• Next, if L is nonsingular, and if we take $F = -L^{-1}\hat{C},\ H = L^{-1}$, then

$$(1.7)\quad (C + DF)(sI - A - BF)^{-1}BH + DH = \hat{T}^{-1}(s) = \begin{bmatrix} p_1^{-1}(s) & & \\ & \ddots & \\ & & p_m^{-1}(s) \end{bmatrix}.$$

However, (i) as addressed in [9], the explicit computation of the matrix L above is ill-conditioned and should be avoided; and (ii) it is pointed out in [23, p. 797, lines 1, 2, 3 below (50.5)] that "the closed-loop eigenvalues are at the assumed stable zeros of $T(s)$ (i.e., $C(sI - A)^{-1}B + D$) and at the selected stable zeros of the polynomials $p_i(s)$." But, if the ith row of D is nonzero for some i, then $l_i = 0$, and $p_i(s) =$ constant, which has no stable zeros, so, if the zeros of $C(sI - A)^{-1}B + D$ are not stable and/or $D \neq 0$, then $A + BF$ is not necessarily stable, and thus, the (F, H) above cannot solve the state feedback decoupling problem with stability for system (1.1) when $D \neq 0$. Hence, the results in [23, 37] also cannot give any direction for establishing a numerically implementable method for solving the state feedback decoupling problem with stability for general proper systems of the form (1.1) with $D \neq 0$.

Based on the above observations, we can conclude that the state feedback decoupling problem with stability for general proper systems of the form (1.1) is still an open problem from both theoretical and numerical points of view. The main purpose of this paper is to develop a numerically reliable method based on a numerical linear algebra technique to solve this open problem.

2. Main results. In this section we will develop a numerical method for solving the state feedback decoupling problem with stability for general proper systems of the form (1.1). Our main idea lies in decomposing the underlying problem into the following two subproblems:

(a) decoupling of a reduced system without a feedthrough matrix, and

(b) simultaneous decoupling and disturbance decoupling of a reduced system with a nonsingular feedthrough term that is further reduced to an "unusual" problem of decoupling.

The obtained solutions are then transformed back to get a solution to the original problem. Our approach is very technical and consists of the following four different stages:

(1) The state feedback decoupling problem with stability for system (1.1) is much more difficult than that for system (1.5). Stage 1 is to eliminate the singularity of matrix D and reduce the underlying problem for system (1.1) into (i) a state feedback decoupling problem with stability for a reduced system without direct feedthrough matrix; and (ii) a simultaneous problem of state feedback decoupling with stability and disturbance decoupling for a reduced system with *nonsingular* direct feedthrough matrix.
(2) The state feedback decoupling problem with stability for a system without direct feedthrough matrix can be solved using the numerical method in [9]; thus, in Stage 2, we consider only the simultaneous problem of state feedback decoupling with stability and disturbance decoupling arising in Stage 1. This simultaneous problem will be reduced to an "unusual" state feedback decoupling problem with stability for a reduced system with *nonsingular* direct feedthrough matrix.
(3) In Stage 3 we first derive a useful reduction property of the "unusual" state feedback decoupling problem with stability produced in Stage 2 and then present a numerically reliable algorithm for solving this "unusual" decoupling problem with stability.
(4) Stage 4 consists of back-transformations of the results in Stages 1, 2, and 3 to the desired solution for the original decoupling problem with stability. An outline of the overall algorithm is given in this stage.

The following auxiliary lemma will be used frequently in this section.

LEMMA 2.1 (cf. [13, 14]). *Given* $E, A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, $C \in \mathbf{R}^{p\times n}$, $D \in \mathbf{R}^{p\times m}$ *with* $\operatorname{rank}(E) = n$,

(i) *the following three statements are equivalent:*

$$\begin{array}{ll} [(a)] & C(sE-A)^{-1}B + D = 0; \\ [(b)] & \max_{s\in\mathbf{C}} \operatorname{rank}\left[\begin{smallmatrix} sE-A & B \\ C & -D \end{smallmatrix}\right] = n; \\ [(c)] & D = 0, \max_{s\in\mathbf{C}} \operatorname{rank}\left[\begin{smallmatrix} sE-A & B \\ C & 0 \end{smallmatrix}\right] = n; \end{array}$$

(ii) *if* $\operatorname{rank}\left[\begin{array}{cc} sE-A & B \end{array}\right] = n$ *for all* $s \in \mathbf{C}$, *then* $\max_{s\in\mathbf{C}} \operatorname{rank}\left[\begin{smallmatrix} sE-A & B \\ C & 0 \end{smallmatrix}\right] = n$ *if and only if* $C = 0$.

2.1. Stage 1—elimination of the singularity of matrix D. The state feedback decoupling problem with stability for systems of the form (1.5) has been studied in [9]. Hence, the results in [9] are used as a bridge to achieve the purpose of Stage 1 in this subsection.

LEMMA 2.2 (cf. [13, 14, 43]). *Given* $A \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{n\times m}, C \in \mathbf{R}^{m\times n}$, *and* $D \in \mathbf{R}^{m\times m}$, *there exist orthogonal matrices* $V_1 \in \mathbf{R}^{n\times n}$ *and* $W \in \mathbf{R}^{m\times m}$ *and a*

permutation matrix $P \in \mathbf{R}^{m\times m}$ *such that*

$$\begin{cases} V_1AV_1^T = \begin{matrix} \\ \\ \\ \end{matrix}\begin{matrix} n_1 & n_2 & n_3 \end{matrix} \\ \qquad\qquad \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \\ \}n_3 \end{matrix}, \\ PCV_1^T = \begin{matrix} \\ \\ \end{matrix}\begin{matrix} n_1 & n_2 & n_3 \end{matrix} \\ \qquad\qquad \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ 0 & C_{22} & C_{23} \end{bmatrix} \begin{matrix} \}m_0 \\ \}m-m_0 \end{matrix}, \end{cases} \quad \begin{cases} V_1BW = \begin{matrix} \\ \\ \\ \end{matrix}\begin{matrix} m_0 & m-m_0 \end{matrix} \\ \qquad\qquad \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ 0 & 0 \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \\ \}n_3 \end{matrix}, \\ PDW = \begin{matrix} \\ \\ \end{matrix}\begin{matrix} m_0 & m-m_0 \end{matrix} \\ \qquad\qquad \begin{bmatrix} D_{11} & 0 \\ D_{21} & 0 \end{bmatrix} \begin{matrix} \}m_0 \\ \}m-m_0 \end{matrix}, \end{cases}$$

(2.1)

where

(2.2)

$$\operatorname{rank}(D_{11}) = m_0,\ \operatorname{rank}\begin{bmatrix} B_{21} & B_{22} \end{bmatrix} = n_2,\ \max_{s\in\mathbf{C}} \operatorname{rank}\begin{bmatrix} -A_{32} & sI-A_{33} \\ C_{22} & C_{23} \end{bmatrix} = n_2+n_3,$$

(2.3)

$$\operatorname{rank}\begin{bmatrix} sI-A_{11} & B_{11} & B_{12} \\ -A_{21} & B_{21} & B_{22} \end{bmatrix} = n_1+n_2 \quad \forall s\in\mathbf{C}.$$

Some important information about the state feedback decoupling for system (1.1) can be read directly from the form (2.1).

LEMMA 2.3. *Given* $A\in\mathbf{R}^{n\times n}, B\in\mathbf{R}^{n\times m}, C\in\mathbf{R}^{m\times n}$, *and* $D\in\mathbf{R}^{m\times m}$, *assume the factorization* (2.1) *has been determined. If there exist* $F\in\mathbf{R}^{m\times n}$ *and* $H\in\mathbf{R}^{m\times m}$ *such that* (1.4) *holds (i.e.,* $(C+DF)(sI-A-BF)^{-1}BH+DH$ *is nonsingular and diagonal), then*

$$D_{21}=0, \quad n_2 = m-m_0 = \operatorname{rank}(B_{22}). \tag{2.4}$$

Proof. First, (2.2) implies that $n_2+n_3 = \max_{s\in\mathbf{C}} \operatorname{rank}\begin{bmatrix} -A_{32} & sI-A_{33} \\ C_{22} & C_{23} \end{bmatrix} \le n_3+(m-m_0)$, so

$$n_2 \le m-m_0. \tag{2.5}$$

Next, note that $\operatorname{rank}\begin{bmatrix} B_{21} & B_{22} \end{bmatrix} = n_2$ and that there exist $\mathcal{F}_{11}\in\mathbf{R}^{m_0\times n_1}$ and $\mathcal{F}_{21}\in\mathbf{R}^{(m-m_0)\times n_1}$ such that

$$A_{21} + \begin{bmatrix} B_{21} & B_{22} \end{bmatrix}\begin{bmatrix} \mathcal{F}_{11} \\ \mathcal{F}_{21} \end{bmatrix} = 0.$$

Let

$$\mathcal{F} = W^TFV_1^T - \begin{bmatrix} \mathcal{F}_{11} & 0 & 0 \\ \mathcal{F}_{21} & 0 & 0 \end{bmatrix}, \qquad \mathcal{H} = W^THP^T,$$

$$\begin{aligned} \mathcal{A} = V_1AV_1^T + (V_1BW)\begin{bmatrix} \mathcal{F}_{11} & 0 & 0 \\ \mathcal{F}_{21} & 0 & 0 \end{bmatrix} &= \begin{bmatrix} A_{11}+\sum_{i=1}^2 B_{1i}\mathcal{F}_{i1} & A_{12} & A_{13} \\ 0 & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix} \\ &=: \begin{bmatrix} A_{11}+\sum_{i=1}^2 B_{1i}\mathcal{F}_{i1} & \mathcal{A}_{12} \\ 0 & \mathcal{A}_{22} \end{bmatrix}, \\ \mathcal{C} = PCV_1^T + PDW\begin{bmatrix} \mathcal{F}_{11} & 0 & 0 \\ \mathcal{F}_{21} & 0 & 0 \end{bmatrix} &= \begin{bmatrix} C_{11}+D_{11}\mathcal{F}_{11} & C_{12} & C_{13} \\ D_{21}\mathcal{F}_{11} & C_{22} & C_{23} \end{bmatrix} \\ &=: \begin{bmatrix} C_{11}+D_{11}\mathcal{F}_{11} & \mathcal{C}_{12} \\ D_{21}\mathcal{F}_{11} & \mathcal{C}_{22} \end{bmatrix}, \end{aligned}$$

$$\mathcal{B} = V_1 BW = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ 0 & 0 \end{bmatrix}, \quad \mathcal{D} = PDW = \begin{bmatrix} D_{11} & 0 \\ D_{21} & 0 \end{bmatrix},$$

and define the integers $\{l_i\}$ and matrix L as follows: If the ith row of the matrix $\mathcal{D}$ is nonzero, this becomes the ith row of the matrix L, and set $l_i = 0$; otherwise, find the lowest positive integer l_i, for which the ith row of $\mathcal{C}\mathcal{A}^{l_i-1}\mathcal{B}$ is nonzero, this then becomes the ith row of the matrix L. It is clear by using $\mathcal{D} = \left[\begin{smallmatrix} D_{11} & 0 \\ D_{21} & 0 \end{smallmatrix}\right]$, where D_{11} is nonsingular, that L is of the form $L = \left[\begin{smallmatrix} D_{11} & 0 \\ L_{21} & L_{22} \end{smallmatrix}\right]$, and if the ith row of $\begin{bmatrix} D_{21} & 0 \end{bmatrix}$ is nonzero, then it is the ith row of $\begin{bmatrix} L_{21} & L_{22} \end{bmatrix}$. Since $(C+DF)(sI-A-BF)^{-1}BH+DH$ is nonsingular and diagonal, and P is a permutation matrix, thus,

$$(\mathcal{C}+\mathcal{D}\mathcal{F})(sI-\mathcal{A}-\mathcal{B}\mathcal{F})^{-1}\mathcal{B}\mathcal{H}+\mathcal{D}\mathcal{H} = P\left((C+DF)(sI-A-BF)^{-1}BH+DH\right)P^T$$

is also nonsingular and diagonal; consequently, by Lemma 1.2, L is nonsingular. Hence, $D_{21} = 0$, and further,

$$L_{22} = \begin{bmatrix} c_{m_0+1}\mathcal{A}^{l_{m_0+1}-1} \\ \vdots \\ c_m\mathcal{A}^{l_m-1} \end{bmatrix} \begin{bmatrix} B_{12} \\ B_{22} \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{c}_{m_0+1}\mathcal{A}_{22}^{l_{m_0+1}-1} \\ \vdots \\ \hat{c}_m\mathcal{A}_{22}^{l_m-1} \end{bmatrix} \begin{bmatrix} B_{22} \\ 0 \end{bmatrix}$$

is nonsingular, where c_{m_0+k} and $\hat{c}_{m_0+k}$ are the kth rows of $\mathcal{C}$ and $\mathcal{C}_{22}$, respectively, $k = 1, \ldots, m-m_0$. Thus,

$$\operatorname{rank}(B_{22}) = \operatorname{rank}\begin{bmatrix} B_{22} \\ 0 \end{bmatrix} = m - m_0, \tag{2.6}$$

which, together with (2.5) and $B_{22} \in \mathbf{R}^{n_2\times(m-m_0)}$, yields that $n_2 = m - m_0 = \operatorname{rank}(B_{22})$. □

Let orthogonal matrix

$$U_1 := \begin{matrix} \\ \end{matrix} \begin{matrix} n_1 & n_2 \end{matrix} \\ \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \end{matrix}$$

be such that

$$U_1 \begin{bmatrix} B_{12} \\ B_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ R_B \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \end{matrix}. \tag{2.7}$$

Define

(2.8)

$$\begin{cases} \tilde{A}_{11} = \begin{bmatrix} U_{11} & U_{12} \end{bmatrix} \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}, \quad \tilde{A}_{12} = \begin{bmatrix} U_{11} & U_{12} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} U_{21}^T \\ U_{22}^T \end{bmatrix}, \\ \tilde{A}_{13} = \begin{bmatrix} U_{11} & U_{12} \end{bmatrix} \begin{bmatrix} A_{13} \\ A_{23} \end{bmatrix}, \quad \tilde{A}_{22} = \begin{bmatrix} A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} U_{21}^T \\ U_{22}^T \end{bmatrix}, \\ \tilde{A}_{32} = \begin{bmatrix} 0 & A_{32} \end{bmatrix} \begin{bmatrix} U_{21}^T \\ U_{22}^T \end{bmatrix}, \\ \tilde{B}_{11} = \begin{bmatrix} U_{11} & U_{12} \end{bmatrix} \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix}, \quad \tilde{C}_{12} = \begin{bmatrix} C_{11} & C_{12} \end{bmatrix} \begin{bmatrix} U_{21}^T \\ U_{22}^T \end{bmatrix}, \\ \tilde{C}_{22} = \begin{bmatrix} 0 & C_{22} \end{bmatrix} \begin{bmatrix} U_{21}^T \\ U_{22}^T \end{bmatrix}, \end{cases}$$

and

$$Q_L = \begin{bmatrix} U_{11} & U_{12} & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, \qquad Q_R = \begin{bmatrix} I & 0 & 0 \\ U_{21} & U_{22} & 0 \\ 0 & 0 & I \end{bmatrix}. \tag{2.9}$$

LEMMA 2.4. *Assume that condition* (2.4) *holds. Then*
(i) U_{11} *and* U_{22} *are nonsingular.*
(ii)

$$\begin{cases} Q_L V_1 A V_1^T Q_R^T = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} \\ A_{21} & \tilde{A}_{22} & A_{23} \\ 0 & \tilde{A}_{32} & A_{33} \end{bmatrix}, \quad Q_L Q_R^T = \begin{bmatrix} U_{11} & 0 & 0 \\ 0 & U_{22}^T & 0 \\ 0 & 0 & I \end{bmatrix}, \\ Q_L V_1 B W = \begin{bmatrix} \tilde{B}_{11} & 0 \\ B_{21} & B_{22} \\ 0 & 0 \end{bmatrix}, \qquad PCV_1^T Q_R^T = \begin{bmatrix} C_{11} & \tilde{C}_{12} & C_{13} \\ 0 & \tilde{C}_{22} & C_{23} \end{bmatrix}, \end{cases} \tag{2.10}$$

and

$$\operatorname{rank} \begin{bmatrix} sU_{11} - A_{11} & \tilde{B}_{11} \end{bmatrix} = n_1 \quad \forall s \in \mathbf{C}, \tag{2.11}$$

$$\max_{s \in \mathbf{C}} \operatorname{rank} \begin{bmatrix} -\tilde{A}_{32} & sI - A_{33} \\ \tilde{C}_{22} & C_{23} \end{bmatrix} = n_2 + n_3. \tag{2.12}$$

Proof. Part (i) has been proved in [1, 5, 10, 19, 32], and part (ii) follows from a simple calculation. □

Transformations Q_L and Q_R have been used in [1, 5, 10, 19, 32], where many numerical examples have shown that such transformations formed based on orthogonal transformations are numerically reliable, in contrast to the (block) Gaussian elimination without pivoting.

Lemmas 2.2 and 2.4 offer a springboard to the objective of this subsection.

THEOREM 2.5. *The state feedback decoupling problem with stability for system* (1.1) *is solvable if and only if the condition* (2.4) *holds as follows:*

$$\operatorname{rank} \begin{bmatrix} \tilde{A}_{32} & sI - A_{33} \end{bmatrix} = n_3, \quad \forall s \in \mathbf{C} \backslash \mathbf{C}^-. \tag{2.13}$$

Furthermore,

(a) *there exist* $F_{22} \in \mathbf{R}^{n_2 \times n_2}$, $F_{23} \in \mathbf{R}^{n_2 \times n_3}$, *and* $H_{22} \in \mathbf{R}^{n_2 \times n_2}$ *solving the following state feedback decoupling problem with stability:*

$$\text{The pencil } \left(\begin{bmatrix} U_{22}^T & 0 \\ 0 & I \end{bmatrix}, \begin{bmatrix} \tilde{A}_{22} + B_{22}F_{22} & A_{23} + B_{22}F_{23} \\ \tilde{A}_{32} & A_{33} \end{bmatrix} \right) \text{ is stable,} \tag{2.14}$$

$$\begin{bmatrix} \tilde{C}_{22} & C_{23} \end{bmatrix} \begin{bmatrix} sU_{22}^T - \tilde{A}_{22} - B_{22}F_{22} & -A_{23} - B_{22}F_{23} \\ -\tilde{A}_{32} & sI - A_{33} \end{bmatrix}^{-1} \begin{bmatrix} B_{22} \\ 0 \end{bmatrix} H_{22} \tag{2.15}$$

is nonsingular and diagonal.

(b) *there exist matrices* $F_{11} \in \mathbf{R}^{m_0\times n_1}$, $F_{12} \in \mathbf{R}^{m_0\times n_2}$, $F_{13} \in \mathbf{R}^{m_0\times n_3}$, *and* $H_{11} \in \mathbf{R}^{m_0\times m_0}$ *solving the following simultaneous problem of state feedback decoupling with stability and disturbance decoupling:*

(2.16) *The pencil* $(U_{11}, \tilde{A}_{11} + \tilde{B}_{11}F_{11})$ *is stable,*

$D_{11}H_{11}+(C_{11}+D_{11}F_{11})(sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11})^{-1}\tilde{B}_{11}H_{11}$ *is nonsingular and diagonal,*
(2.17)

$$\tilde{C}_{12}+D_{11}F_{12}+\begin{bmatrix} C_{11}+D_{11}F_{11}C_{13}+D_{11}F_{13} \end{bmatrix} \times \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} \\ 0 & sI-A_{33} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{A}_{12}+\tilde{B}_{11}F_{12} \\ \tilde{A}_{32} \end{bmatrix} = 0.$$
(2.18)

Proof. Necessity. Let (F, H) solve the state feedback decoupling problem with stability for system (1.1). Then the condition (2.4) follows directly from Lemma 2.3.

With the condition (2.4) we have the factorization (2.10). Denote

$$(2.19)\quad \begin{cases} \begin{bmatrix} I & 0 \\ B_{21} & B_{22} \end{bmatrix} W^T F V_1^T Q_R^T = \begin{bmatrix} I & 0 \\ 0 & B_{22} \end{bmatrix} \begin{matrix} \\ \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \end{bmatrix} \end{matrix} \begin{matrix} \\ \}m_0 \\ \}n_2 \end{matrix}, \\ \begin{bmatrix} I & 0 \\ B_{21} & B_{22} \end{bmatrix} W^T H P^T = \begin{bmatrix} I & 0 \\ 0 & B_{22} \end{bmatrix} \begin{matrix} \\ \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \end{matrix} \begin{matrix} \\ \}m_0 \\ \}n_2 \end{matrix}. \end{cases}$$

As $D_{21} = 0$, we have

$$(2.20)\quad \begin{cases} Q_L V_1 (sI - A - BF) V_1^T Q_R^T \\ \quad = \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{12}-\tilde{B}_{11}F_{12} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} \\ -A_{21}-B_{22}F_{21} & sU_{22}^T-\tilde{A}_{22}-B_{22}F_{22} & -A_{23}-B_{22}F_{23} \\ 0 & -\tilde{A}_{32} & sI-A_{33} \end{bmatrix}, \\ Q_L V_1 B H P^T = \begin{bmatrix} \tilde{B}_{11}H_{11} & \tilde{B}_{11}H_{12} \\ B_{22}H_{21} & B_{22}H_{22} \\ 0 & 0 \end{bmatrix}, \quad P D H P^T = \begin{bmatrix} D_{11}H_{11} & D_{11}H_{12} \\ 0 & 0 \end{bmatrix}, \\ P(C+DF)V_1^T Q_R^T = \begin{bmatrix} C_{11}+D_{11}F_{11} & \tilde{C}_{12}+D_{11}F_{12} & C_{13}+D_{11}F_{13} \\ 0 & \tilde{C}_{22} & C_{23} \end{bmatrix}. \end{cases}$$

Thus, condition (2.13) follows since $A + BF$ is stable.

Note that $DH + (C + DF)(sI - A - BF)^{-1}BH$ is nonsingular and diagonal, P is a permutation matrix, and so

$$P[DH + (C + DF)(sI - A - BF)^{-1}BH]P^T =: \begin{matrix} \\ \begin{bmatrix} T_{11}(s) & T_{12}(s) \\ T_{21}(s) & T_{22}(s) \end{bmatrix} \end{matrix} \begin{matrix} \\ \}m_0 \\ \}n_2 \end{matrix}$$

is also nonsingular and diagonal; i.e.,

$$\begin{cases} T_{12}(s) = D_{11}H_{12} + \begin{bmatrix} C_{11}+D_{11}F_{11} & \tilde{C}_{12}+D_{11}F_{12} & C_{13}+D_{11}F_{13} \end{bmatrix} \\ \qquad \times \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{12}-\tilde{B}_{11}F_{12} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} \\ -A_{21}-B_{22}F_{21} & sU_{22}^T-\tilde{A}_{22}-B_{22}F_{22} & -A_{23}-B_{22}F_{23} \\ 0 & -\tilde{A}_{32} & sI-A_{33} \end{bmatrix}^{-1} \\ \qquad \times \begin{bmatrix} \tilde{B}_{11}H_{12} \\ B_{22}H_{22} \\ 0 \end{bmatrix} = 0, \\ T_{21}(s) = \begin{bmatrix} 0 & \tilde{C}_{22} & C_{23} \end{bmatrix} \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{12}-\tilde{B}_{11}F_{12} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} \\ -A_{21}-B_{22}F_{21} & sU_{22}^T-\tilde{A}_{22}-B_{22}F_{22} & -A_{23}-B_{22}F_{23} \\ 0 & -\tilde{A}_{32} & sI-A_{33} \end{bmatrix}^{-1} \\ \qquad \times \begin{bmatrix} \tilde{B}_{11}H_{11} \\ B_{22}H_{21} \\ 0 \end{bmatrix} = 0, \end{cases} \tag{2.21}$$

$$\begin{aligned} T_{11}(s) = D_{11}H_{11} &+ \begin{bmatrix} C_{11}+D_{11}F_{11} & \tilde{C}_{12}+D_{11}F_{12} & C_{13}+D_{11}F_{13} \end{bmatrix} \\ &\times \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{12}-\tilde{B}_{11}F_{12} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} \\ -A_{21}-B_{22}F_{21} & sU_{22}^T-\tilde{A}_{22}-B_{22}F_{22} & -A_{23}-B_{22}F_{23} \\ 0 & -\tilde{A}_{32} & sI-A_{33} \end{bmatrix}^{-1} \\ &\times \begin{bmatrix} \tilde{B}_{11}H_{11} \\ B_{22}H_{21} \\ 0 \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} T_{22}(s) = \begin{bmatrix} 0 & \tilde{C}_{22} & C_{23} \end{bmatrix} &\times \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{12}-\tilde{B}_{11}F_{12} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} \\ -A_{21}-B_{22}F_{21} & sU_{22}^T-\tilde{A}_{22}-B_{22}F_{22} & -A_{23}-B_{22}F_{23} \\ 0 & -\tilde{A}_{32} & sI-A_{33} \end{bmatrix}^{-1} \\ &\times \begin{bmatrix} \tilde{B}_{11}H_{12} \\ B_{22}H_{22} \\ 0 \end{bmatrix} \end{aligned}$$

are nonsingular and diagonal.

By applying Lemma 2.1(i) to (2.21) we have that

$$\begin{cases} D_{11}H_{12} = 0, \\ n_1+n_2+n_3 = \max\limits_{s\in\mathbf{C}} \operatorname{rank} \\ \times \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{12}-\tilde{B}_{11}F_{12} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} & \tilde{B}_{11}H_{12} \\ -A_{21}-B_{22}F_{21} & sU_{22}^T-\tilde{A}_{22}-B_{22}F_{22} & -A_{23}-B_{22}F_{23} & B_{22}H_{22} \\ 0 & -\tilde{A}_{32} & sI-A_{33} & 0 \\ C_{11}+D_{11}F_{11} & \tilde{C}_{12}+D_{11}F_{12} & C_{13}+D_{11}F_{13} & 0 \end{bmatrix}, \\ n_1+n_2+n_3 = \max\limits_{s\in\mathbf{C}} \operatorname{rank} \\ \times \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{12}-\tilde{B}_{11}F_{12} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} & \tilde{B}_{11}H_{11} \\ -A_{21}-B_{22}F_{21} & sU_{22}^T-\tilde{A}_{22}-B_{22}F_{22} & -A_{23}-B_{22}F_{23} & B_{22}H_{21} \\ 0 & -\tilde{A}_{32} & sI-A_{33} & 0 \\ 0 & \tilde{C}_{22} & C_{23} & 0 \end{bmatrix}, \end{cases}$$

which, together with (2.12) and the nonsingularity of D_{11}, B_{22}, and H, yields that

$$\begin{cases} H_{12}=0,\ H_{11} \text{ and } H_{22} \text{ are nonsingular,} \\ \max\limits_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{12}-\tilde{B}_{11}F_{12} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} \\ 0 & -\tilde{A}_{32} & sI-A_{33} \\ C_{11}+D_{11}F_{11} & \tilde{C}_{12}+D_{11}F_{12} & C_{13}+D_{11}F_{13} \end{bmatrix} = n_1+n_3, \\ \max\limits_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & \tilde{B}_{11}H_{11} \\ -A_{21}-B_{22}F_{21} & B_{22}H_{21} \end{bmatrix} = n_1. \end{cases} \tag{2.22}$$

Since the property (2.11) and the nonsingularity of H_{11} imply that

$$\operatorname{rank} \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & \tilde{B}_{11}H_{11} \end{bmatrix} = \operatorname{rank} \begin{bmatrix} sU_{11}-\tilde{A}_{11} & \tilde{B}_{11} \end{bmatrix} = n_1 \quad \forall s\in\mathbf{C},$$

we have further by applying Lemma 2.1 to (2.22) that

$$\begin{cases} H_{12}=0, \\ \max\limits_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} & -\tilde{A}_{12}-\tilde{B}_{11}F_{12} \\ 0 & sI-A_{33} & -\tilde{A}_{32} \\ C_{11}+D_{11}F_{11} & C_{13}+D_{11}F_{13} & \tilde{C}_{12}+D_{11}F_{12} \end{bmatrix} = n_1+n_3, \\ B_{22}H_{21}=0,\ -A_{21}-B_{22}F_{21}=0, \end{cases}$$

which with the nonsingularity of B_{22} gives that

$$H_{12}=0,\ H_{21}=0, \quad B_{22}F_{21}=-A_{21}, \tag{2.23}$$

and

$$\max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} & \tilde{A}_{12}+\tilde{B}_{11}F_{12} \\ 0 & sI-A_{33} & \tilde{A}_{32} \\ C_{11}+D_{11}F_{11} & C_{13}+D_{11}F_{13} & -(\tilde{C}_{12}+D_{11}F_{12}) \end{bmatrix} = n_1+n_3. \tag{2.24}$$

Consequently, (2.24) and Lemma 2.1(i) give (2.18). Moreover, we obtain

$$\begin{aligned} &Q_L V_1 (sI-A-BF) V_1^T Q_R^T \\ &= \begin{bmatrix} sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11} & -\tilde{A}_{12}-\tilde{B}_{11}F_{12} & -\tilde{A}_{13}-\tilde{B}_{11}F_{13} \\ 0 & sU_{22}^T-\tilde{A}_{22}-B_{22}F_{22} & -A_{23}-B_{22}F_{23} \\ 0 & -\tilde{A}_{32} & sI-A_{33} \end{bmatrix}, \end{aligned}$$

$$\begin{cases} T_{11}(s) = D_{11}H_{11} + (C_{11}+D_{11}F_{11})(sU_{11}-\tilde{A}_{11}-\tilde{B}_{11}F_{11})^{-1}\tilde{B}_{11}H_{11}, \\ T_{22}(s) = \begin{bmatrix} \tilde{C}_{22} & C_{23} \end{bmatrix} \begin{bmatrix} sU_{22}^T-\tilde{A}_{22}-B_{22}F_{22} & -A_{23}-B_{22}F_{23} \\ -\tilde{A}_{32} & sI-A_{33} \end{bmatrix}^{-1} \begin{bmatrix} B_{22} \\ 0 \end{bmatrix} H_{22}. \end{cases}$$

Now, $A+BF$ is stable, and $T_{11}(s)$ and $T_{22}(s)$ are nonsingular and diagonal. Hence, (2.14)–(2.17) hold.

Sufficiency. Assume that the conditions (2.4) and (2.13) hold, and that F_{11}, F_{12}, F_{13}, F_{22}, F_{23}, H_{11} and H_{22} satisfy (2.14)–(2.18). Define F_{21}, H_{12}, and H_{21} by (2.23) and determine F and H by (2.19). Then a simple calculation yields that the matrix $A+BF$ is stable, and that $(C+DF)(sI-A-BF)^{-1}BH+DH$ is nonsingular and diagonal. Hence, (F,H) solves the state feedback decoupling problem with stability for the system (1.1). □

Remark 2.6. Problem (2.18) is called a disturbance decoupling problem [39].

The feedthrough matrices in the state feedback decoupling problems (2.15) and (2.17) are zero and nonsingular, respectively. By this feature the problems (2.15) and (2.17) are easier to solve than the original problem (1.4), as shown in the next subsections.

2.2. Stage 2—rejection of the disturbance in (2.5)–(2.18). Since the problem of (2.14) and (2.15) has been solved in [9], in this subsection we shall study only the simultaneous problem of (2.5), (2.17), and (2.18). We will reject the disturbance and consequently transform the simultaneous problem of (2.5), (2.17), and (2.18) into a single problem like that of (2.5) and (2.17).

LEMMA 2.7 (cf. [27, 30, 43]). *Assume that the factorizations* (2.1) *and* (2.10) *have been determined. There exist orthogonal matrices* $U_2 \in \mathbf{R}^{(n_1+n_3+m_0)\times(n_1+n_3+m_0)}$ *and* $V_2 \in \mathbf{R}^{(n_1+n_3)\times(n_1+n_3)}$ *such that*

$$U_2 \left[\begin{array}{ccc|cc} -D_{11} & -C_{11} & -C_{13} & \tilde{C}_{12} & I \\ -\tilde{B}_{11} & sU_{11}-\tilde{A}_{11} & -\tilde{A}_{13} & \tilde{A}_{12} & 0 \\ 0 & 0 & sI-A_{33} & \tilde{A}_{32} & 0 \end{array}\right] \left[\begin{array}{ccc} I_{m_0} & 0 & 0 \\ 0 & V_2 & 0 \\ 0 & 0 & I \end{array}\right]^T$$

$$= \begin{array}{c} \begin{array}{ccccccc} m_0 & n_1+n_3-\tau-\nu & \tau & \nu & n_2 & m_0 & \end{array} \\ \left[\begin{array}{ccccccc} \mathcal{D}_{11} & \star & \star & \star & \star & \star & \}m_0 \\ 0 & s\tilde{\Theta}_{11}-\tilde{\Phi}_{11} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12} & s\tilde{\Theta}_{13}-\tilde{\Phi}_{13} & \Delta & \tilde{\Psi} & \}n_1+n_3-\tau-\nu \\ 0 & 0 & s\Theta-\Phi & s\tilde{\Theta}_{23}-\tilde{\Phi}_{23} & 0 & \Psi & \}\tau \\ 0 & 0 & 0 & s\tilde{\Theta}_{33}-\tilde{\Phi}_{33} & 0 & 0 & \}\nu \end{array}\right] \end{array}, \tag{2.25}$$

where $\star$ *denotes subblocks that we are not interested in;* $\tilde{\Theta}_{11}$, Θ, *and* $\tilde{\Theta}_{33}$ *are nonsingular; and*

$$\operatorname{rank}\left[\begin{array}{cc} s\tilde{\Theta}_{11}-\tilde{\Phi}_{11} & \Delta \end{array}\right] = n_1+n_3-\tau-\nu \quad \forall s \in \mathbf{C}, \tag{2.26}$$

$$\operatorname{rank}\left[\begin{array}{cc} s\Theta-\Phi & \Psi \end{array}\right] = \tau \quad \forall s \in \mathbf{C}. \tag{2.27}$$

Proof. By computing the QR factorization of $\left[\begin{smallmatrix} D_{11} \\ \tilde{B}_{11} \end{smallmatrix}\right]$ first and then computing the controllability staircase form [27, 30, 43] of the pair

$$\left(\mathcal{N}^T \left[\begin{array}{ccc} -D_{11} & -C_{11} & -C_{13} \\ -\tilde{B}_{11} & sU_{11}-\tilde{A}_{11} & -\tilde{A}_{13} \\ 0 & 0 & sI-A_{33} \end{array}\right] \left[\begin{array}{c} 0 \\ I_{n_1+n_3} \end{array}\right]\right), \quad \left(\mathcal{N}^T \left[\begin{array}{cc} \tilde{C}_{12} & I \\ \tilde{A}_{12} & 0 \\ \tilde{A}_{32} & 0 \end{array}\right]\right),$$

we get the factorization (2.25). Here $\mathcal{N}$ is a column orthogonal matrix whose columns span the null space of $\left[\begin{smallmatrix} D_{11} \\ \tilde{B}_{11} \end{smallmatrix}\right]^T$. ∎

THEOREM 2.8. *Assume that the conditions* (2.4) *and* (2.13) *hold and the factorization* (2.25) *has been determined. Then the simultaneous problem of* (2.5), (2.17), *and* (2.18) *is solvable if and only if*

$$\sigma(\tilde{\Theta}_{11},\ \tilde{\Phi}_{11}) \backslash \mathbf{C}^- = \sigma(A_{33}) \backslash \mathbf{C}^-, \tag{2.28}$$

and there exists a $\mathcal{K} \in \mathbf{R}^{m_0\times\tau}$ *satisfying the following "unusual" state feedback decoupling with stability:*

(2.29) *The pencil* $(\Theta, \Phi+\Psi\mathcal{K})$ *is stable,* $\mathcal{K}(s\Theta-\Phi-\Psi\mathcal{K})^{-1}\Psi$ *is diagonal.*

Moreover, if (2.28) *and* (2.29) *hold with* $\mathcal{K} \in \mathbf{R}^{m_0\times\tau}$, *then* (2.5), (2.17), *and* (2.18) *hold with* F_{11}, F_{12}, F_{13}, *and* H_{11} *determined by*

$$D_{11}\begin{bmatrix} F_{11} & F_{13}\end{bmatrix} = -\begin{bmatrix} 0 & \mathcal{K} & \hat{\mathcal{K}}\end{bmatrix}V_2 - \begin{bmatrix} C_{11} & C_{13}\end{bmatrix}, \quad D_{11}F_{12} = -\tilde{C}_{12},$$

$$D_{11}H_{11} \text{ is nonsingular and diagonal,}$$

where $\hat{\mathcal{K}} \in \mathbf{R}^{m_0\times\nu}$ *is arbitrary.*

In the above, for any square matrices N *and* M, $\sigma(N,M)$ *denotes the finite spectrum of the pencil* (N,M), *and* $\sigma(M) = \sigma(I,M)$.

Proof. First, the factorization (2.25) and the properties (2.11) and (2.13) yield

$$\begin{aligned}
&\operatorname{rank}\begin{bmatrix} \mathcal{D}_{11} & \star & \star & \star & \star & \star \\ 0 & s\tilde{\Theta}_{11}-\tilde{\Phi}_{11} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12} & s\tilde{\Theta}_{13}-\tilde{\Phi}_{13} & \Delta & \tilde{\Psi} \\ 0 & 0 & s\Theta-\Phi & s\tilde{\Theta}_{23}-\tilde{\Phi}_{23} & 0 & \Psi \\ 0 & 0 & 0 & s\tilde{\Theta}_{33}-\tilde{\Phi}_{33} & 0 & 0 \end{bmatrix} \\
&= \operatorname{rank}\begin{bmatrix} -D_{11} & -C_{11} & -C_{13} & \tilde{C}_{12} & I \\ -\tilde{B}_{11} & sU_{11}-\tilde{A}_{11} & -\tilde{A}_{13} & \tilde{A}_{12} & 0 \\ 0 & 0 & sI-A_{33} & \tilde{A}_{32} & 0 \end{bmatrix} = n_1+n_3+m_0 \quad \forall s \in \mathbf{C}\backslash\mathbf{C}^-,
\end{aligned}$$

and thus, the pencil $(\tilde{\Theta}_{33}, \tilde{\Phi}_{33})$ is stable.

Next, the property (2.27) holds, so there exists a $\mathcal{K}_0$ such that the pencil $(\Theta, \Phi + \Psi\mathcal{K}_0)$ is stable [39]. Let

$$\begin{bmatrix} \mathcal{F}_1^{(0)} & \mathcal{F}_3^{(0)}\end{bmatrix} = -\begin{bmatrix} \overset{n_1+n_3-\tau-\nu}{0} & \overset{\tau}{\mathcal{K}_0} & \overset{\nu}{0}\end{bmatrix}V_2.$$

Then

$$\begin{aligned}
&U_2\begin{bmatrix} -D_{11} & -C_{11}+\mathcal{F}_1^{(0)} & -C_{13}+\mathcal{F}_3^{(0)} \\ -\tilde{B}_{11} & sU_{11}-\tilde{A}_{11} & -\tilde{A}_{13} \\ 0 & 0 & sI-A_{33} \end{bmatrix}\begin{bmatrix} I_{m_0} & 0 \\ 0 & V_2 \end{bmatrix}^T \\
&= \begin{bmatrix} \mathcal{D}_{11} & \star & \star & \star \\ 0 & s\tilde{\Theta}_{11}-\tilde{\Phi}_{11} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12} & s\tilde{\Theta}_{13}-\tilde{\Phi}_{13} \\ 0 & 0 & s\Theta-\Phi & s\tilde{\Theta}_{23}-\tilde{\Phi}_{23} \\ 0 & 0 & 0 & s\tilde{\Theta}_{33}-\tilde{\Phi}_{33} \end{bmatrix} - \begin{bmatrix} \star \\ \tilde{\Psi} \\ \Psi \\ 0 \end{bmatrix}\begin{bmatrix} 0 & 0 & \mathcal{K}_0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \mathcal{D}_{11} & \star & \star & \star \\ 0 & s\tilde{\Theta}_{11}-\tilde{\Phi}_{11} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12}-\tilde{\Psi}\mathcal{K}_0 & s\tilde{\Theta}_{13}-\tilde{\Phi}_{13} \\ 0 & 0 & s\Theta-\Phi-\Psi\mathcal{K}_0 & s\tilde{\Theta}_{23}-\tilde{\Phi}_{23} \\ 0 & 0 & 0 & s\tilde{\Theta}_{33}-\tilde{\Phi}_{33} \end{bmatrix},
\end{aligned}$$

so, by using the stability of the pencils $(\Theta, \Phi+\Psi\mathcal{K}_0)$ and $(\tilde{\Theta}_{33}, \tilde{\Phi}_{33})$ we obtain

(2.30)

$$\begin{aligned}
\sigma(A_{33})\backslash\mathbf{C}^- &\subseteq \sigma\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & U_{11} & 0 \\ 0 & 0 & I \end{bmatrix}, \begin{bmatrix} D_{11} & C_{11}-\mathcal{F}_1^{(0)} & C_{13}-\mathcal{F}_3^{(0)} \\ \tilde{B}_{11} & \tilde{A}_{11} & \tilde{A}_{13} \\ 0 & 0 & A_{33} \end{bmatrix}\right)\backslash\mathbf{C}^- \\
&= \sigma\left(\begin{bmatrix} \tilde{\Theta}_{11} & \tilde{\Theta}_{12} & \tilde{\Theta}_{13} \\ 0 & \Theta & \tilde{\Theta}_{23} \\ 0 & 0 & \tilde{\Theta}_{33} \end{bmatrix}, \begin{bmatrix} \tilde{\Phi}_{11} & \tilde{\Phi}_{12}+\tilde{\Psi}\mathcal{K}_0 & \tilde{\Phi}_{13} \\ 0 & \Phi+\Psi\mathcal{K}_0 & \tilde{\Phi}_{23} \\ 0 & 0 & \tilde{\Phi}_{33} \end{bmatrix}\right)\backslash\mathbf{C}^- \\
&= \sigma(\tilde{\Theta}_{11}, \tilde{\Phi}_{11})\backslash\mathbf{C}^-.
\end{aligned}$$

For any $F_{1i} \in \mathbf{R}^{m_0 \times n_i}$ $(i = 1, 2, 3)$ and $H_{11} \in \mathbf{R}^{m_0 \times m_0}$, define

$$\begin{cases} \mathcal{F}_1 = C_{11} + D_{11}F_{11}, \quad \mathcal{F}_2 = \tilde{C}_{12} + D_{11}F_{12}, \quad \mathcal{F}_3 = C_{13} + D_{11}F_{13}, \quad \mathcal{H} = D_{11}H_{11}, \\ \qquad\qquad\qquad\qquad n_1 + n_3 - \tau - \nu \quad \tau \quad \nu \\ \left[\begin{array}{cc} \mathcal{F}_1 & \mathcal{F}_3 \end{array}\right] V_2^T = -\left[\begin{array}{ccc} \tilde{\mathcal{K}} & \mathcal{K} & \hat{\mathcal{K}} \end{array}\right]. \end{cases}$$

We have that

$$U_2 \left[\begin{array}{ccc} -D_{11} & -C_{11} + \mathcal{F}_1 & -C_{13} + \mathcal{F}_3 \\ -\tilde{B}_{11} & sU_{11} - \tilde{A}_{11} & -\tilde{A}_{13} \\ 0 & 0 & sI - A_{33} \end{array}\right] \left[\begin{array}{cc} I_{m_0} & 0 \\ 0 & V_2 \end{array}\right]^T$$

$$= \left[\begin{array}{cccc} \mathcal{D}_{11} & \star & \star & \star \\ 0 & s\tilde{\Theta}_{11} - \tilde{\Phi}_{11} - \tilde{\Psi}\tilde{\mathcal{K}} & s\tilde{\Theta}_{12} - \tilde{\Phi}_{12} - \tilde{\Psi}\mathcal{K} & s\tilde{\Theta}_{13} - \tilde{\Phi}_{13} - \tilde{\Psi}\hat{\mathcal{K}} \\ 0 & -\Psi\tilde{\mathcal{K}} & s\Theta - \Phi - \Psi\mathcal{K} & s\tilde{\Theta}_{23} - \tilde{\Phi}_{23} - \Psi\hat{\mathcal{K}} \\ 0 & 0 & 0 & s\tilde{\Theta}_{33} - \tilde{\Phi}_{33} \end{array}\right],$$

$$\begin{aligned} &\sigma\ (U_{11},\ \tilde{A}_{11} + \tilde{B}_{11}F_{11}) = \sigma(U_{11},\ \tilde{A}_{11} + \tilde{B}_{11}D_{11}^{-1}(\mathcal{F}_1 - C_{11})) \\ &= \sigma\left(\left[\begin{array}{cc} 0 & 0 \\ 0 & U_{11} \end{array}\right], \left[\begin{array}{cc} D_{11} & C_{11} - \mathcal{F}_1 \\ \tilde{B}_{11} & \tilde{A}_{11} \end{array}\right]\right) \\ &= \sigma\left(\left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & U_{11} & 0 \\ 0 & 0 & I \end{array}\right], \left[\begin{array}{ccc} D_{11} & C_{11} - \mathcal{F}_1 & C_{13} - \mathcal{F}_3 \\ \tilde{B}_{11} & \tilde{A}_{11} & \tilde{A}_{13} \\ 0 & 0 & A_{33} \end{array}\right]\right) \backslash \sigma(A_{33}) \\ &= \sigma\left(\left[\begin{array}{cccc} 0 & \star & \star & \star \\ 0 & \tilde{\Theta}_{11} & \tilde{\Theta}_{12} & \tilde{\Theta}_{13} \\ 0 & 0 & \Theta & \tilde{\Theta}_{23} \\ 0 & 0 & 0 & \tilde{\Theta}_{33} \end{array}\right], \left[\begin{array}{cccc} -\mathcal{D}_{11} & \star & \star & \star \\ 0 & \tilde{\Phi}_{11} + \tilde{\Psi}\tilde{\mathcal{K}} & \tilde{\Phi}_{12} + \tilde{\Psi}\mathcal{K} & \tilde{\Phi}_{13} + \tilde{\Psi}\hat{\mathcal{K}} \\ 0 & \Psi\tilde{\mathcal{K}} & \Phi + \Psi\mathcal{K} & \tilde{\Phi}_{23} + \Psi\hat{\mathcal{K}} \\ 0 & 0 & 0 & \tilde{\Phi}_{33} \end{array}\right]\right) \backslash \sigma(A_{33}), \end{aligned}$$

$$\begin{aligned} &D_{11}H_{11} + (C_{11} + D_{11}F_{11})(sU_{11} - \tilde{A}_{11} - \tilde{B}_{11}F_{11})^{-1}\tilde{B}_{11}H_{11} \\ &= \mathcal{H} + \mathcal{F}_1(sU_{11} - \tilde{A}_{11} - \tilde{B}_{11}D_{11}^{-1}(\mathcal{F}_1 - C_{11}))^{-1}\tilde{B}_{11}D_{11}^{-1}\mathcal{H} \\ &= \mathcal{H} + \left[\begin{array}{ccc} 0 & \mathcal{F}_1 & \mathcal{F}_3 \end{array}\right] \left[\begin{array}{ccc} -D_{11} & -C_{11} + \mathcal{F}_1 & -C_{13} + \mathcal{F}_3 \\ -\tilde{B}_{11} & sU_{11} - \tilde{A}_{11} & -\tilde{A}_{13} \\ 0 & 0 & sI - A_{33} \end{array}\right]^{-1} \left[\begin{array}{c} -\mathcal{H} \\ 0 \\ 0 \end{array}\right] \\ &= \mathcal{H} + \left[\begin{array}{ccc} \tilde{\mathcal{K}} & \mathcal{K} & \hat{\mathcal{K}} \end{array}\right] \left[\begin{array}{ccc} s\tilde{\Theta}_{11} - \tilde{\Phi}_{11} - \tilde{\Psi}\tilde{\mathcal{K}} & s\tilde{\Theta}_{12} - \tilde{\Phi}_{12} - \tilde{\Psi}\mathcal{K} & s\tilde{\Theta}_{13} - \tilde{\Phi}_{13} - \tilde{\Psi}\hat{\mathcal{K}} \\ -\Psi\tilde{\mathcal{K}} & s\Theta - \Phi - \Psi\mathcal{K} & s\tilde{\Theta}_{23} - \tilde{\Phi}_{23} - \Psi\hat{\mathcal{K}} \\ 0 & 0 & s\tilde{\Theta}_{33} - \tilde{\Phi}_{33} \end{array}\right]^{-1} \\ &\qquad \times \left[\begin{array}{c} \tilde{\Psi}\mathcal{H} \\ \Psi\mathcal{H} \\ 0 \end{array}\right], \end{aligned}$$

and

$$\begin{aligned} &\tilde{C}_{12} + D_{11}F_{12} + \left[\begin{array}{cc} C_{11} + D_{11}F_{11} & C_{13} + D_{11}F_{13} \end{array}\right] \\ &\quad \times \left[\begin{array}{cc} sU_{11} - \tilde{A}_{11} - \tilde{B}_{11}F_{11} - \tilde{A}_{13} - \tilde{B}_{11}F_{13} & \\ 0 & sI - A_{33} \end{array}\right]^{-1} \left[\begin{array}{c} \tilde{A}_{12} + \tilde{B}_{11}F_{12} \\ \tilde{A}_{32} \end{array}\right] \\ &= \mathcal{F}_2 + \left[\begin{array}{cc} \mathcal{F}_1 & \mathcal{F}_3 \end{array}\right] \left[\begin{array}{cc} sU_{11} - \tilde{A}_{11} - \tilde{B}_{11}D_{11}^{-1}(\mathcal{F}_1 - C_{11}) & -\tilde{A}_{13} - \tilde{B}_{11}D_{11}^{-1}(\mathcal{F}_3 - C_{13}) \\ 0 & sI - A_{33} \end{array}\right]^{-1} \\ &\qquad \times \left[\begin{array}{c} \tilde{A}_{12} + \tilde{B}_{11}D_{11}^{-1}(\mathcal{F}_2 - \tilde{C}_{12}) \\ \tilde{A}_{32} \end{array}\right] \end{aligned}$$

$$= \mathcal{F}_2 + \begin{bmatrix} 0 & \mathcal{F}_1 & \mathcal{F}_3 \end{bmatrix} \begin{bmatrix} -D_{11} & -C_{11}+\mathcal{F}_1 & -C_{13}+\mathcal{F}_3 \\ -\tilde{B}_{11} & sU_{11}-\tilde{A}_{11} & -\tilde{A}_{13} \\ 0 & 0 & sI-A_{33} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{C}_{12}-\mathcal{F}_2 \\ \tilde{A}_{12} \\ \tilde{A}_{32} \end{bmatrix}$$

$$= \mathcal{F}_2 + \begin{bmatrix} \tilde{\mathcal{K}} & \mathcal{K} & \hat{\mathcal{K}} \end{bmatrix} \begin{bmatrix} s\tilde{\Theta}_{11}-\tilde{\Phi}_{11}-\tilde{\Psi}\tilde{\mathcal{K}} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12}-\tilde{\Psi}\mathcal{K} & s\tilde{\Theta}_{13}-\tilde{\Phi}_{13}-\tilde{\Psi}\hat{\mathcal{K}} \\ -\Psi\tilde{\mathcal{K}} & s\Theta-\Phi-\Psi\mathcal{K} & s\tilde{\Theta}_{23}-\tilde{\Phi}_{23}-\Psi\hat{\mathcal{K}} \\ 0 & 0 & s\tilde{\Theta}_{33}-\tilde{\Phi}_{33} \end{bmatrix}^{-1} \times \begin{bmatrix} \tilde{\Psi}\mathcal{F}_2-\Delta \\ \Psi\mathcal{F}_2 \\ 0 \end{bmatrix}.$$

As a result, we get by using Lemma 2.1, factorization (2.25), and the stability of the pencil $(\tilde{\Theta}_{33}, \tilde{\Phi}_{33})$ that (2.5)–(2.18) hold as follows:

$$\Longleftrightarrow \begin{cases} \sigma\left(\begin{bmatrix} \tilde{\Theta}_{11} & \tilde{\Theta}_{12} & \tilde{\Theta}_{13} \\ 0 & \Theta & \tilde{\Theta}_{23} \\ 0 & 0 & \tilde{\Theta}_{33} \end{bmatrix}, \begin{bmatrix} \tilde{\Phi}_{11}+\tilde{\Psi}\tilde{\mathcal{K}} & \tilde{\Phi}_{12}+\tilde{\Psi}\mathcal{K} & \tilde{\Phi}_{13}+\tilde{\Psi}\hat{\mathcal{K}} \\ \Psi\tilde{\mathcal{K}} & \Phi+\Psi\mathcal{K} & \tilde{\Phi}_{23}+\Psi\hat{\mathcal{K}} \\ 0 & 0 & \tilde{\Phi}_{33} \end{bmatrix}\right) \backslash \mathbf{C}^- = \sigma(A_{33}) \backslash \mathbf{C}^-, \\ \mathcal{H} + \begin{bmatrix} \tilde{\mathcal{K}} & \mathcal{K} & \hat{\mathcal{K}} \end{bmatrix} \begin{bmatrix} s\tilde{\Theta}_{11}-\tilde{\Phi}_{11}-\tilde{\Psi}\tilde{\mathcal{K}} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12}-\tilde{\Psi}\mathcal{K} & s\tilde{\Theta}_{13}-\tilde{\Phi}_{13}-\tilde{\Psi}\hat{\mathcal{K}} \\ -\Psi\tilde{\mathcal{K}} & s\Theta-\Phi-\Psi\mathcal{K} & s\tilde{\Theta}_{23}-\tilde{\Phi}_{23}-\Psi\hat{\mathcal{K}} \\ 0 & 0 & s\tilde{\Theta}_{33}-\tilde{\Phi}_{33} \end{bmatrix}^{-1} \\ \qquad \times \begin{bmatrix} \tilde{\Psi}\mathcal{H} \\ \Psi\mathcal{H} \\ 0 \end{bmatrix} \text{ is nonsingular and diagonal}, \\ \mathcal{F}_2 + \begin{bmatrix} \tilde{\mathcal{K}} & \mathcal{K} & \hat{\mathcal{K}} \end{bmatrix} \begin{bmatrix} s\tilde{\Theta}_{11}-\tilde{\Phi}_{11}-\tilde{\Psi}\tilde{\mathcal{K}} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12}-\tilde{\Psi}\mathcal{K} & s\tilde{\Theta}_{13}-\tilde{\Phi}_{13}-\tilde{\Psi}\hat{\mathcal{K}} \\ -\Psi\tilde{\mathcal{K}} & s\Theta-\Phi-\Psi\mathcal{K} & s\tilde{\Theta}_{23}-\tilde{\Phi}_{23}-\Psi\hat{\mathcal{K}} \\ 0 & 0 & s\tilde{\Theta}_{33}-\tilde{\Phi}_{33} \end{bmatrix}^{-1} \\ \qquad \times \begin{bmatrix} \tilde{\Psi}\mathcal{F}_2-\Delta \\ \Psi\mathcal{F}_2 \\ 0 \end{bmatrix} = 0 \end{cases}$$

$$\Longleftrightarrow \begin{cases} \sigma\left(\begin{bmatrix} \tilde{\Theta}_{11} & \tilde{\Theta}_{12} \\ 0 & \Theta \end{bmatrix}, \begin{bmatrix} \tilde{\Phi}_{11}+\tilde{\Psi}\tilde{\mathcal{K}} & \tilde{\Phi}_{12}+\tilde{\Psi}\mathcal{K} \\ \Psi\tilde{\mathcal{K}} & \Phi+\Psi\mathcal{K} \end{bmatrix}\right) \backslash \mathbf{C}^- = \sigma(A_{33}) \backslash \mathbf{C}^-, \\ \mathcal{H} \text{ is nonsingular and diagonal}, \\ \begin{bmatrix} \tilde{\mathcal{K}} & \mathcal{K} \end{bmatrix} \begin{bmatrix} s\tilde{\Theta}_{11}-\tilde{\Phi}_{11}-\tilde{\Psi}\tilde{\mathcal{K}} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12}-\tilde{\Psi}\mathcal{K} \\ -\Psi\tilde{\mathcal{K}} & s\Theta-\Phi-\Psi\mathcal{K} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{\Psi} \\ \Psi \end{bmatrix} \text{ is diagonal}, \\ \mathcal{F}_2 = 0, \quad \begin{bmatrix} \tilde{\mathcal{K}} & \mathcal{K} \end{bmatrix} \begin{bmatrix} s\tilde{\Theta}_{11}-\tilde{\Phi}_{11}-\tilde{\Psi}\tilde{\mathcal{K}} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12}-\tilde{\Psi}\mathcal{K} \\ -\Psi\tilde{\mathcal{K}} & s\Theta-\Phi-\Psi\mathcal{K} \end{bmatrix}^{-1} \begin{bmatrix} \Delta \\ 0 \end{bmatrix} = 0. \end{cases} \tag{2.31}$$

Because Lemma 2.1 implies that

$$\begin{bmatrix} \tilde{\mathcal{K}} & \mathcal{K} \end{bmatrix} \begin{bmatrix} s\tilde{\Theta}_{11}-\tilde{\Phi}_{11}-\tilde{\Psi}\tilde{\mathcal{K}} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12}-\tilde{\Psi}\mathcal{K} \\ -\Psi\tilde{\mathcal{K}} & s\Theta-\Phi-\Psi\mathcal{K} \end{bmatrix}^{-1} \begin{bmatrix} \Delta \\ 0 \end{bmatrix} = 0$$

$$\Longleftrightarrow \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} s\tilde{\Theta}_{11}-\tilde{\Phi}_{11}-\tilde{\Psi}\tilde{\mathcal{K}} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12}-\tilde{\Psi}\mathcal{K} & \Delta \\ -\Psi\tilde{\mathcal{K}} & s\Theta-\Phi-\Psi\mathcal{K} & 0 \\ \tilde{\mathcal{K}} & \mathcal{K} & 0 \end{bmatrix} = n_1+n_3-\nu$$

$$\Longleftrightarrow \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} s\tilde{\Theta}_{11}-\tilde{\Phi}_{11} & s\tilde{\Theta}_{12}-\tilde{\Phi}_{12} & \Delta \\ 0 & s\Theta-\Phi & 0 \\ \tilde{\mathcal{K}} & \mathcal{K} & 0 \end{bmatrix} = n_1+n_3-\nu$$

$$\iff \max_{s\in\mathbf{C}} \operatorname{rank}\begin{bmatrix} s\tilde{\Theta}_{11}-\tilde{\Phi}_{11} & \Delta \\ \tilde{\mathcal{K}} & 0 \end{bmatrix} = (n_1+n_3)-\tau-\nu$$

(since $\Theta \in \mathbf{R}^{\tau\times\tau}$ is nonsingular and thus $\max_{s\in\mathbf{C}} \operatorname{rank}(s\Theta-\Phi)=\tau$)

$$\iff \tilde{\mathcal{K}} = 0 \qquad \text{(since (2.26) holds)},$$

we obtain by using (2.31) that

$$(2.5)-(2.18)\text{ hold} \iff \begin{cases} \mathcal{F}_2 = 0, \quad \tilde{\mathcal{K}} = 0, \\ \mathcal{K}(s\Theta-\Phi-\Psi\mathcal{K})^{-1}\Psi \text{ is diagonal}, \\ (\sigma(\tilde{\Theta}_{11},\ \tilde{\Phi}_{11})\cup\sigma(\Theta,\ \Phi+\Psi\mathcal{K}))\backslash\mathbf{C}^- = \sigma(A_{33})\backslash\mathbf{C}^-, \end{cases}$$

$$\iff \begin{cases} \mathcal{F}_2 = 0, \quad \tilde{\mathcal{K}} = 0, \\ \mathcal{K}(s\Theta-\Phi-\Psi\mathcal{K})^{-1}\Psi \text{ is diagonal}, \\ \sigma(\tilde{\Theta}_{11},\ \tilde{\Phi}_{11})\backslash\mathbf{C}^- = \sigma(A_{33})\backslash\mathbf{C}^-,\ \sigma(\Theta,\ \Phi+\Psi\mathcal{K}) \subseteq \mathbf{C}^- \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(since (2.30) holds)} \end{cases}$$

$$\iff \begin{cases} \mathcal{F}_2 = 0, \quad \tilde{\mathcal{K}} = 0, \\ \mathcal{K}(s\Theta-\Phi-\Psi\mathcal{K})^{-1}\Psi \text{ is diagonal}, \\ \sigma(\tilde{\Theta}_{11},\ \tilde{\Phi}_{11})\backslash\mathbf{C}^- = \sigma(A_{33})\backslash\mathbf{C}^-,\ \text{pencil } (\Theta, \Phi+\Psi\mathcal{K}) \text{ is stable}. \end{cases}$$

Hence, Theorem 2.8 holds. □

2.3. Stage 3—a numerical solution to Problem (2.29). The main difference between the problem (2.29) and the usual state feedback decoupling problem is that the problem (2.29) does not require the nonsingularity of the transfer function $\mathcal{K}(s\Theta-\Phi-\Psi\mathcal{K})^{-1}\Psi$. To our knowledge, the problem (2.29) has not been studied yet, and our attempts to characterize its solvability conditions by extending the results in [24, 25, 26, 39] have failed. We will develop a numerically reliable algorithm in this subsection for solving the problem (2.29).

The following result is trivial.

COROLLARY 2.9. *Assume that the factorization* (2.25) *has been determined and that* $m_0 = 1$*; then the problem* (2.29) *is always solvable and all its solutions are given by all matrices* $\mathcal{K}$ *satisfying that the pencil* $(\Theta, \Phi+\Psi\mathcal{K})$ *is stable.*

In the following we consider the case that $m_0 > 1$.

THEOREM 2.10. *Assume that the factorization* (2.25) *has been determined and that* $m_0 > 1$*. There exist nonnegative integers* τ_i *(*$i = 1, \ldots, 5$*), a permutation matrix* $\mathcal{P} \in \mathbf{R}^{m_0\times m_0}$*, and orthogonal matrices* $\mathcal{U}_1, \mathcal{U}_2, \mathcal{V}_1, \mathcal{V}_2 \in \mathbf{R}^{\tau\times\tau}$*,* $\mathcal{U}_3, \mathcal{V}_3 \in \mathbf{R}^{(\tau_1+\tau_2+\tau_3)\times(\tau_1+\tau_2+\tau_3)}$ *with partitioning,*

$$\mathcal{U}_2 = \begin{array}{c} \begin{matrix} \sum_{i=1}^3 \tau_i & \tau_4+\tau_5 \end{matrix} \\ \begin{bmatrix} \mathcal{U}_{11} & \mathcal{U}_{12} \\ \mathcal{U}_{21} & \mathcal{U}_{22} \end{bmatrix} \begin{matrix} \}\sum_{i=1}^3 \tau_i \\ \}\tau_4+\tau_5 \end{matrix} \end{array}, \quad \mathcal{V}_2 = \begin{array}{c} \begin{matrix} \sum_{i=1}^3 \tau_i & \tau_4+\tau_5 \end{matrix} \\ \begin{bmatrix} \mathcal{V}_{11} & \mathcal{V}_{12} \\ \mathcal{V}_{21} & \mathcal{V}_{22} \end{bmatrix} \begin{matrix} \}\sum_{i=1}^3 \tau_i \\ \}\tau_4+\tau_5 \end{matrix} \end{array},$$

such that $\operatorname{rank}(\mathcal{U}_{11}) = \sum_{i=1}^{3} \tau_i$, $\operatorname{rank}(\mathcal{V}_{22}) = \tau_4 + \tau_5$, *and*

$$
\begin{cases}
\begin{bmatrix} \mathcal{U}_3 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathcal{U}_{11} & \mathcal{U}_{12} \\ 0 & I \end{bmatrix} \mathcal{U}_1 (s\Theta - \Phi) \left(\begin{bmatrix} \mathcal{V}_3 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ \mathcal{V}_{21} & \mathcal{V}_{22} \end{bmatrix} \mathcal{V}_1 \right)^T \\
\qquad = \begin{array}{c} \begin{matrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 & \tau_5 \end{matrix} \\ \begin{bmatrix} s\Theta_{11} - \Phi_{11} & 0 & 0 & 0 & 0 \\ s\Theta_{21} - \Phi_{21} & s\Theta_{22} - \Phi_{22} & 0 & 0 & 0 \\ s\Theta_{31} - \Phi_{31} & s\Theta_{32} - \Phi_{32} & s\Theta_{33} - \Phi_{33} & 0 & -\Phi_{35} \\ 0 & 0 & 0 & s\Theta_{44} - \Phi_{44} & s\Theta_{45} - \Phi_{45} \\ 0 & 0 & 0 & s\Theta_{54} - \Phi_{54} & s\Theta_{55} - \Phi_{55} \end{bmatrix} \end{array} \begin{matrix} \}\tau_1 \\ \}\tau_2 \\ \}\tau_3 \\ \}\tau_5 \\ \}\tau_4 \end{matrix}, \\
\begin{bmatrix} \mathcal{U}_3 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathcal{U}_{11} & \mathcal{U}_{12} \\ 0 & I \end{bmatrix} \mathcal{U}_1 \Psi \mathcal{P}^T = \begin{array}{c} \begin{matrix} m_1 & \tau_5 & m_0 - m_1 - \tau_5 \end{matrix} \\ \begin{bmatrix} \Psi_{11} & 0 & 0 \\ \Psi_{21} & 0 & \Psi_{23} \\ \Psi_{31} & 0 & \Psi_{33} \\ 0 & \Psi_{42} & \Psi_{43} \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{matrix} \}\tau_1 \\ \}\tau_2 \\ \}\tau_3 \\ \}\tau_5 \\ \}\tau_4 \end{matrix},
\end{cases}
\tag{2.32}
$$

where

$$
0 < m_1 < m_0, \quad 0 \le \tau_5 \le 1, \tag{2.33}
$$

$$
\text{if } \tau_5 = 0, \text{ then } \tau_3 = \tau_4 = 0, \tag{2.34}
$$

(2.35)

$$
\operatorname{rank} \begin{bmatrix} s\Theta_{11} - \Phi_{11} & 0 & 0 & \Psi_{11} \\ s\Theta_{21} - \Phi_{21} & s\Theta_{22} - \Phi_{22} & 0 & \Psi_{21} \\ s\Theta_{31} - \Phi_{31} & s\Theta_{32} - \Phi_{32} & s\Theta_{33} - \Phi_{33} & \Psi_{31} \end{bmatrix} = \tau_1 + \tau_2 + \tau_3 \quad \forall s \in \mathbf{C},
$$

(2.36)

$$
\operatorname{rank} \begin{bmatrix} s\Theta_{22} - \Phi_{22} & \Psi_{23} \end{bmatrix} = \tau_2 \quad \forall s \in \mathbf{C},
$$

and furthermore, if $\tau_5 = 1$, *we also have*

$$
\Psi_{42} \neq 0, \tag{2.37}
$$

$$
\operatorname{rank} \begin{bmatrix} s\Theta_{33} - \Phi_{33} & \Phi_{35} \end{bmatrix} = \tau_3, \quad \operatorname{rank}(s\Theta_{54} - \Phi_{54}) = \tau_4 \quad \forall s \in \mathbf{C}. \tag{2.38}
$$

Proof. The constructive proof of Theorem 2.10 is given in [3]. □

Similarly to the factorization (2.10), the factorization (2.32) is also numerically reliable.

We are now ready to derive a useful reduction property of the problem (2.29) using the factorization (2.32).

THEOREM 2.11. *Assume that factorizations* (2.25) *and* (2.32) *have been determined. Then the problem* (2.29) *is solvable if and only if*

$$
\text{pencils } (\Theta_{22}, \Phi_{22}) \text{ and } (\Theta_{33}, \Phi_{33}) \text{ are stable}, \tag{2.39}
$$

$$
\text{the pencil } \left(\begin{bmatrix} \Theta_{44} & \Theta_{45} \\ \Theta_{54} & \Theta_{55} \end{bmatrix}, \begin{bmatrix} \Phi_{44} & \Phi_{45} \\ \Phi_{54} & \Phi_{44} \end{bmatrix} \right) \text{ is stable if } \Psi_{43} \neq 0, \tag{2.40}
$$

and there exists a matrix $\mathcal{K}_{11} \in \mathbf{R}^{m_1 \times \tau_1}$ *such that*

(2.41)
$$\text{pencil } (\Theta_{11}, \Phi_{11} + \Psi_{11}\mathcal{K}_{11}) \text{ is stable}, \quad \mathcal{K}_{11}(s\Theta_{11} - \Phi_{11} - \Psi_{11}\mathcal{K}_{11})^{-1}\Psi_{11} \text{ is diagonal.}$$

Moreover, if (2.39), (2.40), *and* (2.41) *hold, then one solution* $\mathcal{K}$ *of the problem* (2.29) *is given by*

$$\mathcal{K}\left(\begin{bmatrix} \mathcal{V}_3 & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} I & 0 \\ \mathcal{V}_{21} & \mathcal{V}_{22} \end{bmatrix}\mathcal{V}_1\right)^T = \mathcal{P}^T \begin{array}{c} \begin{matrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 & \tau_5 \end{matrix} \\ \begin{bmatrix} \mathcal{K}_{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathcal{K}_{24} & \mathcal{K}_{25} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{matrix} \}m_1 \\ \}\tau_5 \\ \}m_0 - m_1 - \tau_5, \end{matrix}$$

(2.42)

where

(2.43)
$$\begin{bmatrix} \mathcal{K}_{24} & \mathcal{K}_{25} \end{bmatrix} = 0 \text{ if } \Psi_{43} \neq 0,$$

(2.44)
$$\text{the pencil } \left(\begin{bmatrix} \Theta_{44} & \Theta_{45} \\ \Theta_{54} & \Theta_{55} \end{bmatrix}, \begin{bmatrix} \Phi_{44} & \Phi_{45} \\ \Phi_{54} & \Phi_{55} \end{bmatrix} + \begin{bmatrix} \Psi_{42} \\ 0 \end{bmatrix}\begin{bmatrix} \mathcal{K}_{24} & \mathcal{K}_{25} \end{bmatrix}\right)$$
is stable if $\Psi_{43} = 0$.

Proof. Denote

$$\mathcal{Q}_{\mathcal{L}} = \begin{bmatrix} \mathcal{U}_3 & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \mathcal{U}_{11} & \mathcal{U}_{12} \\ 0 & I \end{bmatrix}\mathcal{U}_1, \quad \mathcal{Q}_{\mathcal{R}} = \begin{bmatrix} \mathcal{V}_3 & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} I & 0 \\ \mathcal{V}_{21} & \mathcal{V}_{22} \end{bmatrix}\mathcal{V}_1,$$

(2.45)
$$\mathcal{Q}_{\mathcal{L}}\Psi\mathcal{P}^T =: \begin{array}{c} \begin{matrix} m_1 & \tau_5 & m_0 - m_1 - \tau_5 \end{matrix} \\ \begin{bmatrix} \Psi_1 & \Psi_2 & \Psi_3 \end{bmatrix} \end{array},$$
$$\text{i.e., } \Psi_1 := \begin{bmatrix} \Psi_{11} \\ \Psi_{21} \\ \Psi_{31} \\ 0 \\ 0 \end{bmatrix}, \ \Psi_2 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ \Psi_{42} \\ 0 \end{bmatrix}, \ \Psi_3 := \begin{bmatrix} 0 \\ \Psi_{23} \\ \Psi_{33} \\ \Psi_{43} \\ 0 \end{bmatrix}.$$

For any $\mathcal{K} \in \mathbf{R}^{m_0 \times \tau}$, let

$$\mathcal{P}\mathcal{K}\mathcal{Q}_{\mathcal{R}}^T =: \begin{array}{c} \begin{matrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 & \tau_5 \end{matrix} \\ \begin{bmatrix} \mathcal{K}_{11} & \mathcal{K}_{12} & \mathcal{K}_{13} & \mathcal{K}_{14} & \mathcal{K}_{15} \\ \mathcal{K}_{21} & \mathcal{K}_{22} & \mathcal{K}_{23} & \mathcal{K}_{24} & \mathcal{K}_{25} \\ \mathcal{K}_{31} & \mathcal{K}_{32} & \mathcal{K}_{33} & \mathcal{K}_{34} & \mathcal{K}_{35} \end{bmatrix} \end{array} \begin{matrix} \}m_1 \\ \}\tau_5 \\ \}m_0 - m_1 - \tau_5 \end{matrix} = \begin{bmatrix} \mathcal{K}_1 \\ \mathcal{K}_2 \\ \mathcal{K}_3 \end{bmatrix} \begin{matrix} \}m_1 \\ \}\tau_5 \\ \}m_0 - m_1 - \tau_5. \end{matrix}$$

Necessity. Assume that (2.29) holds with $\mathcal{K} \in \mathbf{R}^{m_0 \times \tau}$. Since $\mathcal{K}(s\Theta - \Phi - \Psi\mathcal{K})^{-1}\Psi$ is diagonal, so $\mathcal{P}\mathcal{K}\mathcal{Q}_{\mathcal{R}}^T(\mathcal{Q}_{\mathcal{L}}(s\Theta - \Phi)\mathcal{Q}_{\mathcal{R}}^T - \mathcal{Q}_{\mathcal{L}}\Psi\mathcal{P}^T\mathcal{P}\mathcal{K}\mathcal{Q}_{\mathcal{R}}^T)^{-1}\mathcal{Q}_{\mathcal{L}}\Psi\mathcal{P}^T$ is also diagonal; i.e.,

$$\begin{bmatrix} \mathcal{K}_1 \\ \mathcal{K}_2 \\ \mathcal{K}_3 \end{bmatrix}(\mathcal{Q}_{\mathcal{L}}(s\Theta - \Phi)\mathcal{Q}_{\mathcal{R}}^T - \Psi_1\mathcal{K}_1 - \Psi_2\mathcal{K}_2 - \Psi_3\mathcal{K}_3)^{-1}\begin{bmatrix} \Psi_1 & \Psi_2 & \Psi_3 \end{bmatrix} \text{ is diagonal.}$$

Thus, we have

$$\begin{aligned}
&\mathcal{K}_3(\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T-\Psi_1\mathcal{K}_1-\Psi_2\mathcal{K}_2-\Psi_3\mathcal{K}_3)^{-1}\begin{bmatrix}\Psi_1 & \Psi_2\end{bmatrix}=0,\\
&\mathcal{K}_1(\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T-\Psi_1\mathcal{K}_1-\Psi_2\mathcal{K}_2-\Psi_3\mathcal{K}_3)^{-1}\begin{bmatrix}\Psi_2 & \Psi_3\end{bmatrix}=0,\\
&\mathcal{K}_2(\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T-\Psi_1\mathcal{K}_1-\Psi_2\mathcal{K}_2-\Psi_3\mathcal{K}_3)^{-1}\Psi_1=0,\\
&\mathcal{K}_2(\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T-\Psi_1\mathcal{K}_1-\Psi_2\mathcal{K}_2-\Psi_3\mathcal{K}_3)^{-1}\Psi_3=0.
\end{aligned}$$

By using Lemma 2.1(i) we obtain

$$\begin{aligned}
\tau&=\max_{s\in\mathbf{C}}\operatorname{rank}\begin{bmatrix}\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T-\Psi_1\mathcal{K}_1-\Psi_2\mathcal{K}_2-\Psi_3\mathcal{K}_3 & \Psi_1 & \Psi_2\\ \mathcal{K}_3 & 0 & 0\end{bmatrix}\\
&=\max_{s\in\mathbf{C}}\operatorname{rank}\begin{bmatrix}\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T & \Psi_1 & \Psi_2\\ \mathcal{K}_3 & 0 & 0\end{bmatrix},
\end{aligned}\tag{2.46}$$

$$\begin{aligned}
\tau&=\max_{s\in\mathbf{C}}\operatorname{rank}\begin{bmatrix}\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T-\Psi_1\mathcal{K}_1-\Psi_2\mathcal{K}_2-\Psi_3\mathcal{K}_3 & \Psi_2 & \Psi_3\\ \mathcal{K}_1 & 0 & 0\end{bmatrix}\\
&=\max_{s\in\mathbf{C}}\operatorname{rank}\begin{bmatrix}\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T & \Psi_2 & \Psi_3\\ \mathcal{K}_1 & 0 & 0\end{bmatrix}\\
&=\max_{s\in\mathbf{C}}\operatorname{rank}\begin{bmatrix}
s\Theta_{11}-\Phi_{11} & 0 & 0 & 0 & 0 & 0 & 0\\
s\Theta_{21}-\Phi_{21} & s\Theta_{22}-\Phi_{22} & 0 & 0 & 0 & 0 & \Psi_{23}\\
s\Theta_{31}-\Phi_{31} & s\Theta_{32}-\Phi_{32} & s\Theta_{33}-\Phi_{33} & 0 & -\Phi_{35} & 0 & \Psi_{33}\\
0 & 0 & 0 & s\Theta_{44}-\Phi_{44} & s\Theta_{45}-\Phi_{45} & \Psi_{42} & \Psi_{43}\\
0 & 0 & 0 & s\Theta_{54}-\Phi_{54} & s\Theta_{55}-\Phi_{55} & 0 & 0\\
\mathcal{K}_{11} & \mathcal{K}_{12} & \mathcal{K}_{13} & \mathcal{K}_{14} & \mathcal{K}_{15} & 0 & 0
\end{bmatrix},
\end{aligned}\tag{2.47}$$

$$\begin{aligned}
\tau&=\max_{s\in\mathbf{C}}\operatorname{rank}\begin{bmatrix}\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T-\Psi_1\mathcal{K}_1-\Psi_2\mathcal{K}_2-\Psi_3\mathcal{K}_3 & \Psi_1\\ \mathcal{K}_2 & 0\end{bmatrix}\\
&=\max_{s\in\mathbf{C}}\operatorname{rank}\begin{bmatrix}\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T-\Psi_3\mathcal{K}_3 & \Psi_1\\ \mathcal{K}_2 & 0\end{bmatrix},
\end{aligned}\tag{2.48}$$

$$\begin{aligned}
\tau&=\max_{s\in\mathbf{C}}\operatorname{rank}\begin{bmatrix}\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T-\Psi_1\mathcal{K}_1-\Psi_2\mathcal{K}_2-\Psi_1\mathcal{K}_1 & \Psi_3\\ \mathcal{K}_2 & 0\end{bmatrix}\\
&=\max_{s\in\mathbf{C}}\operatorname{rank}\begin{bmatrix}\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T-\Psi_1\mathcal{K}_1 & \Psi_3\\ \mathcal{K}_2 & 0\end{bmatrix}.
\end{aligned}\tag{2.49}$$

In (2.32), we have

$$\operatorname{rank}\begin{bmatrix}\mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T & \Psi_1 & \Psi_2\end{bmatrix}=\tau\quad\forall s\in\mathbf{C},$$

which, with (2.46) and Lemma 2.1(ii), gives that

$$\mathcal{K}_3=0,\ \text{i.e.,}\ \begin{bmatrix}\mathcal{K}_{31} & \mathcal{K}_{32} & \mathcal{K}_{33} & \mathcal{K}_{34} & \mathcal{K}_{35}\end{bmatrix}=0.\tag{2.50}$$

Note that

$$\operatorname{rank}\begin{bmatrix}
s\Theta_{22}-\Phi_{22} & 0 & 0 & 0 & 0 & \Psi_{23}\\
s\Theta_{32}-\Phi_{32} & s\Theta_{33}-\Phi_{33} & 0 & -\Phi_{35} & 0 & \Psi_{33}\\
0 & 0 & s\Theta_{44}-\Phi_{44} & s\Theta_{45}-\Phi_{45} & \Psi_{42} & \Psi_{43}\\
0 & 0 & s\Theta_{54}-\Phi_{54} & s\Theta_{55}-\Phi_{55} & 0 & 0
\end{bmatrix}=\tau-\tau_1\quad\forall s\in\mathbf{C},$$

and (2.47) yields that

$$\max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} s\Theta_{22}-\Phi_{22} & 0 & 0 & 0 & 0 & \Psi_{23} \\ s\Theta_{32}-\Phi_{32} & s\Theta_{33}-\Phi_{33} & 0 & -\Phi_{35} & 0 & \Psi_{33} \\ 0 & 0 & s\Theta_{44}-\Phi_{44} & s\Theta_{45}-\Phi_{45} & \Psi_{42} & \Psi_{43} \\ 0 & 0 & s\Theta_{54}-\Phi_{54} & s\Theta_{55}-\Phi_{55} & 0 & 0 \\ \mathcal{K}_{12} & \mathcal{K}_{13} & \mathcal{K}_{14} & \mathcal{K}_{15} & 0 & 0 \end{bmatrix} = \tau-\tau_1;$$

thus, it follows from Lemma 2.1(ii) that

$$\begin{bmatrix} \mathcal{K}_{12} & \mathcal{K}_{13} & \mathcal{K}_{14} & \mathcal{K}_{15} \end{bmatrix} = 0. \tag{2.51}$$

Next, (2.50) is true, and (2.48) is reduced to

$$\begin{aligned} \tau &= \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} \mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T & \Psi_1 \\ \mathcal{K}_2 & 0 \end{bmatrix} \\ &= \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} s\Theta_{11}-\Phi_{11} & 0 & 0 & 0 & 0 & \Psi_{11} \\ s\Theta_{21}-\Phi_{21} & s\Theta_{22}-\Phi_{22} & 0 & 0 & 0 & \Psi_{21} \\ s\Theta_{31}-\Phi_{31} & s\Theta_{32}-\Phi_{32} & s\Theta_{33}-\Phi_{33} & 0 & -\Phi_{35} & \Psi_{31} \\ 0 & 0 & 0 & s\Theta_{44}-\Phi_{44} & s\Theta_{45}-\Phi_{45} & 0 \\ 0 & 0 & 0 & s\Theta_{54}-\Phi_{54} & s\Theta_{55}-\Phi_{55} & 0 \\ \mathcal{K}_{21} & \mathcal{K}_{22} & \mathcal{K}_{23} & \mathcal{K}_{24} & \mathcal{K}_{25} & 0 \end{bmatrix}, \end{aligned}$$

i.e.,

$$\max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} s\Theta_{11}-\Phi_{11} & 0 & 0 & \Psi_{11} \\ s\Theta_{21}-\Phi_{21} & s\Theta_{22}-\Phi_{22} & 0 & \Psi_{21} \\ s\Theta_{31}-\Phi_{31} & s\Theta_{32}-\Phi_{32} & s\Theta_{33}-\Phi_{33} & \Psi_{31} \\ \mathcal{K}_{21} & \mathcal{K}_{22} & \mathcal{K}_{23} & 0 \end{bmatrix} = \tau_1+\tau_2+\tau_3,$$

which, with (2.36) and Lemma 2.1(ii), gives

$$\begin{bmatrix} \mathcal{K}_{21} & \mathcal{K}_{22} & \mathcal{K}_{23} \end{bmatrix} = 0. \tag{2.52}$$

Now, (2.51) and (2.52) hold, so (2.49) is reduced to

$$\begin{aligned} \tau &= \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} \mathcal{Q}_{\mathcal{L}}(s\Theta-\Phi)\mathcal{Q}_{\mathcal{R}}^T-\Psi_1\mathcal{K}_1 & \Psi_3 \\ \mathcal{K}_2 & 0 \end{bmatrix} \\ &= \max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} s\Theta_{11}-\Phi_{11}-\Psi_{11}\mathcal{K}_{11} & 0 & 0 & 0 & 0 & 0 \\ s\Theta_{21}-\Phi_{21}-\Psi_{21}\mathcal{K}_{11} & s\Theta_{22}-\Phi_{22} & 0 & 0 & 0 & \Psi_{23} \\ s\Theta_{31}-\Phi_{31}-\Psi_{31}\mathcal{K}_{11} & s\Theta_{32}-\Phi_{32} & s\Theta_{33}-\Phi_{33} & 0 & -\Phi_{35} & \Psi_{33} \\ 0 & 0 & 0 & s\Theta_{44}-\Phi_{44} & s\Theta_{45}-\Phi_{45} & \Psi_{43} \\ 0 & 0 & 0 & s\Theta_{54}-\Phi_{54} & s\Theta_{55}-\Phi_{55} & 0 \\ 0 & 0 & 0 & \mathcal{K}_{24} & \mathcal{K}_{25} & 0 \end{bmatrix}, \end{aligned}$$

which gives

$$\max_{s\in\mathbf{C}} \operatorname{rank} \begin{bmatrix} s\Theta_{44}-\Phi_{44} & s\Theta_{45}-\Phi_{45} & \Psi_{43} \\ s\Theta_{54}-\Phi_{54} & s\Theta_{55}-\Phi_{55} & 0 \\ \mathcal{K}_{24} & \mathcal{K}_{25} & 0 \end{bmatrix} = \tau_4+\tau_5. \tag{2.53}$$

If $\Psi_{43}=0$, then (2.53) is always true. Otherwise, if $\Psi_{43}\neq 0$, then $\tau_5=1$ and

$$\operatorname{rank} \begin{bmatrix} s\Theta_{44}-\Phi_{44} & s\Theta_{45}-\Phi_{45} & \Psi_{43} \\ s\Theta_{54}-\Phi_{54} & s\Theta_{55}-\Phi_{55} & 0 \end{bmatrix} = \tau_4+\tau_5 \quad \forall s\in\mathbf{C}. \tag{2.54}$$

In this case, it follows from (2.53), (2.54), and Lemma 2.1(ii) that

$$\begin{bmatrix} \mathcal{K}_{24} & \mathcal{K}_{25} \end{bmatrix} = 0. \tag{2.55}$$

We have shown above that (2.42) and (2.43) hold. Therefore, (2.39), (2.40), and (2.41) follow directly from (2.50), (2.51), (2.52), and (2.55).

Sufficiency. If $\Psi_{43} = 0$, then, since

$$\operatorname{rank} \begin{bmatrix} s\Theta_{44} - \Phi_{44} & s\Theta_{45} - \Phi_{45} & \Psi_{42} \\ s\Theta_{54} - \Phi_{54} & s\Theta_{55} - \Phi_{55} & 0 \end{bmatrix} = \tau_4 + \tau_5 \quad \forall s \in \mathbf{C},$$

there always exist matrices $\mathcal{K}_{24}$ and $\mathcal{K}_{25}$ such that (2.44) holds [30]. Let $\mathcal{K} \in \mathbf{R}^{m_0 \times \tau}$ be given by (2.42), (2.43), and (2.44). Because (2.39), (2.40), and (2.41) hold, a simple calculation yields that the pencil $(\Theta, \Phi + \Psi\mathcal{K})$ is stable, and $\mathcal{P}\mathcal{K}\mathcal{Q}_{\mathcal{R}}^T(\mathcal{Q}_{\mathcal{L}}(s\Theta - \Phi)\mathcal{Q}_{\mathcal{R}}^T - \mathcal{Q}_{\mathcal{L}}\Psi\mathcal{P}^T\mathcal{P}\mathcal{K}\mathcal{Q}_{\mathcal{R}}^T)^{-1}\mathcal{Q}_{\mathcal{L}}\Psi\mathcal{P}^T$ is diagonal, i.e., $\mathcal{K}(s\Theta - \Phi - \Psi\mathcal{K})^{-1}\Psi$ is diagonal. □

Theorem 2.11 and Corollary 2.9 lead to the following algorithm, which is based only on orthogonal transformations and solutions of some linear systems of equations for solving the problem (2.29).

ALGORITHM 1.

Input: $\Theta, \Phi \in \mathbf{R}^{\tau \times \tau}$, $\Psi \in \mathbf{R}^{\tau \times m_0}$ *satisfying* (2.27).

Output: $\mathcal{K} \in \mathbf{R}^{m_0 \times \tau}$ *(if possible) solving the problem* (2.29).

Step 0. *Set* $K := \emptyset$, $M := I_{m_0}$, $N := I_\tau$, $l := 0$.

Step 1. *If* $m_0 = 1$, *compute* $\mathcal{K}$ *such that the pencil* $(\Theta, \Phi + \Psi\mathcal{K})$ *is stable, and set* $K := \left[\begin{smallmatrix} \mathcal{K} & 0 \\ 0 & K \end{smallmatrix}\right]$, *then go to Step* 3. *Otherwise, if* $m_0 > 1$, *go to Step* 2.

Step 2. *Compute the factorization* (2.32). *If* (2.39) *or* (2.40) *fails, print "The problem* (2.29) *is not solvable" and stop. Otherwise, compute* $\begin{bmatrix} \mathcal{K}_{24} & \mathcal{K}_{25} \end{bmatrix}$ *based on* (2.43) *and* (2.44). *Set*

$$K := \begin{array}{c} \begin{matrix} \tau_2 & \tau_3 & \tau_4 & \tau_5 & \end{matrix} \\ \begin{bmatrix} 0 & 0 & \mathcal{K}_{24} & \mathcal{K}_{25} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K \end{bmatrix} \end{array} \begin{array}{l} \}\tau_5 \\ \}m_0 - m_1 - \tau_5 \\ l \end{array},$$

$$M := M \begin{bmatrix} \mathcal{P}^T & 0 \\ 0 & I \end{bmatrix}, \quad N := N \begin{bmatrix} \mathcal{Q}_{\mathcal{R}}^T & 0 \\ 0 & I \end{bmatrix},$$

$$\Theta := \Theta_{11}, \ \Phi := \Phi_{11}, \ \Psi := \Psi_{11}, \ l = l + m_0 - m_1, \ \tau = \tau_1, \ m_0 := m_1.$$

Go to Step 1.

Step 3. *Compute* $\mathcal{K}$ *by solving the linear system*

$$\mathcal{K}N = MK. \tag{2.56}$$

Output $\mathcal{K}$.

2.4. Stage 4—an overall algorithm. The results in subsections 2.1, 2.2, and 2.3 can be combined to provide an overall algorithm for solving the state feedback decoupling problem with stability for general proper systems of the form (1.1) as follows.

ALGORITHM 2.

Input: $A \in \mathbf{R}^{n \times n}$, $B \in \mathbf{R}^{n \times m}$, $C \in \mathbf{R}^{m \times n}$, *and* $D \in \mathbf{R}^{m \times m}$.

Output: *Matrices* $F \in \mathbf{R}^{m\times n}$ *and* $H \in \mathbf{R}^{m\times m}$ *(if possible) such that* $A+BF$ *is stable and* (1.4) *holds.*

Step 1. *Compute the factorization* (2.1)*. If the condition* (2.4) *is not true, print "The studied problem is not solvable" and stop. Otherwise, compute the factorization* (2.10)*. If the condition* (2.13) *is not true, print "The studied problem is not solvable" and stop.*

Step 2. *Solve the problem of* (2.14) *and* (2.15) *by the algorithm in* [9]*. If it is not solvable, print "The studied problem is not solvable" and stop. Otherwise, compute* $(\begin{bmatrix} F_{22} & F_{23} \end{bmatrix}, H_{22})$ *such that* (2.14) *and* (2.15) *hold.*

Step 3. *Compute the staircase form* (2.25)*. If the condition* (2.28) *does not hold, print "The studied problem is not solvable" and stop. Otherwise, perform Algorithm* 1 *without Step* 3 *to get matrices* M*,* N*, and* K*.*

Step 4. *Solve* (2.56) *to get* $\mathcal{K}$ *and then solve the following linear systems of equations to get* F *and* H*:*

$$\begin{bmatrix} D_{11} & 0 \\ B_{21} & B_{22} \end{bmatrix} W^T F V_1^T Q_R^T = -\begin{bmatrix} C_{11}+E_1 & \tilde{C}_{12} & C_{13}+E_3 \\ A_{21} & -B_{22}F_{22} & -B_{22}F_{23} \end{bmatrix},$$

$$\begin{bmatrix} D_{11} & 0 \\ B_{21} & B_{22} \end{bmatrix} W^T H P^T = \begin{bmatrix} \Lambda & 0 \\ 0 & B_{22}H_{22} \end{bmatrix},$$

where Λ *is an arbitrary nonsingular and diagonal matrix, and*

$$\begin{bmatrix} E_1 & E_3 \end{bmatrix} = \begin{bmatrix} n_1+n_3-\tau-\nu & \tau & \nu \\ 0 & \mathcal{K} & 0 \end{bmatrix} \mathcal{V}_2.$$

Output F *and* H*.*

Note that Algorithm 2 has the following features:

- The factorizations (2.1), (2.10), (2.25), and (2.32) are numerically reliable.[1]
- Steps 1 and 3 are implemented based on only orthogonal transformations, in which the condition (2.13) is equivalent to that the pair $(A_{33}, \tilde{A}_{32})$ is stabilizable and thus it can be verified easily [43].
- Step 2 is implemented based on the algorithm in [9] which is numerically reliable.
- Linear systems of equations in Steps 3 and 4 can be solved efficiently by the SVD method [22].

Hence, Algorithm 2 can be implemented in a numerically reliable manner.

In the following, we apply Algorithm 2 to an example generated by MATLAB 7.0.

EXAMPLE 2.12. *Let*

$$A(:,1:3) = \begin{bmatrix} -134.7151815770 & 684.5200688479 & -252.4858402416 \\ 2965.6910031615 & 2968.6795856120 & 3127.3190640174 \\ 2496.6636879933 & 5313.7625190612 & 2258.0418588501 \\ -224.6817995288 & 1986.6127660631 & -532.3798160083 \\ -37.1081164625 & -3118.1270598251 & 369.5187260686 \\ 1410.6906971850 & 2447.1024041232 & 1348.3759688211 \end{bmatrix},$$

[1] The computations of the factorizations (2.1), (2.10), (2.25), and (2.32) require the usage of the notion *numerical rank* of a matrix. The most reliable method for deciding the rank of a matrix $M \in \mathbf{R}^{m\times n}$, $m \geq n$, is as follows: Compute the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ of M, as in [27], and consider a singular value σ_j to be zero if $\sigma_j \leq \epsilon\sigma_1$, where ϵ bounds the relative error in M. The number of remaining nonzero singular values is then taken to be the (numerical) rank of the matrix.

$$A(:,\,4\,:\,6) = \begin{bmatrix} 316.6433509627 & 300.0280484275 & -10.467634805 \\ 1276.6140059526 & 1011.4552055403 & 820.772696295 \\ 2365.7472100096 & 2039.6108213012 & 784.077193552 \\ 916.7671233048 & 857.0012773342 & 10.752404856 \\ -1430.5834820660 & -1313.1753655886 & -112.512294456 \\ 1080.7385716405 & 918.0951249252 & 424.352681385 \end{bmatrix},$$

$$B = \begin{bmatrix} 425.29217463199 & -80.88171998499 & -145.72982759769 \\ 3982.20690197082 & 3016.72330542203 & 3062.57237309851 \\ 5436.32609118648 & 2736.92286843590 & 2560.48573537041 \\ 1335.34779570743 & -74.51223868009 & -247.07435777624 \\ -2331.54633119938 & -255.18364603148 & -20.55480073218 \\ 2660.00572592472 & 1506.16763608974 & 1448.73127764291 \end{bmatrix},$$

$$C(:,\,1\,:\,3) = \begin{bmatrix} 0.4447033643532 & 0.9218129707448 & 0.4057062130621 \\ 0.6154323481001 & 0.7382072458107 & 0.9354696991076 \\ 0.7919370374270 & 0.1762661444946 & 0.9169044399134 \end{bmatrix},$$

$$C(:,\,4\,:\,6) = \begin{bmatrix} 0.4102702069910 & 0.3528681322170 & 0.1388908819570 \\ 0.8936495309135 & 0.8131664973038 & 0.2027652185603 \\ 0.0578913047843 & 0.0098613006609 & 0.1987217426615 \end{bmatrix},$$

$$D = \begin{bmatrix} 0.95012928514718 & 0.48598246870930 & 0.45646766516834 \\ 0.23113851357429 & 0.89129896614890 & 0.01850364324822 \\ 0.60684258354179 & 0.76209683302739 & 0.82140716429525 \end{bmatrix}.$$

By using Algorithm 2, *we get a solution* (F, H) *of the state feedback decoupling problem with stability for the proper system described by the* (A, B, C, D) *quadruple as follows:*

$$F(:,\,1\,:\,3) = \begin{bmatrix} 13.213245245439 & 6.533314996605 & -7.120728173265 \\ -5.071573120781 & -1.923440437567 & 0.236601849708 \\ 45.978278670010 & -28.856343706120 & 26.995864311313 \end{bmatrix},$$

$$F(:,\,4\,:\,6) = \begin{bmatrix} -5.266769113169 & -7.409085032473 & -3.479842456999 \\ 1.574115104850 & 1.054592476902 & 0.489990507618 \\ -58.329243480873 & -2.194096577547 & 8.908103088096 \end{bmatrix},$$

$$H = \begin{bmatrix} 1.67404446726824 & -0.11964439680952 & -0.92759516260175 \\ -0.41647243581381 & 1.17375817197623 & 0.20499869641804 \\ -0.85035677245799 & -1.00061468473081 & 1.71251901466682 \end{bmatrix}.$$

It has been verified that $A + BF$ *is stable with eigenvalues*

$$\sigma(A + BF) = \{-2368.44594880864, -1583.30208404670, -0.46317761699,$$

$$-0.00497367650, -0.00764132113, -0.11009271581\},$$

and $DH + (C + DF)(sI - A - BF)^{-1}BH$ *is nonsingular. It has also been verified that* $DH + (C + DF)(sI - A - BF)^{-1}BH$ *is diagonal as follows:*

- *Compute the transfer function* $T_f(s) = DH + (C + DF)(sI - A - BF)^{-1}BH$ *by* MATLAB 7.0

  $$T_f(s) = tf(\mathrm{ss}(A + B * F, B * H, C + D * F, D * H)).$$

  Let $(T_f(s))_{diag}$ *be a diagonal matrix that satisfies that all diagonal elements of* $T_f(s) - (T_f(s))_{diag}$ *are zeros.*

- *Calculate the peak gain of the frequency responses (as measured by the largest singular value) of* $T_f(s)$ *and* $T_f(s) - (T_f(s))_{diag}$ *using the code* $\mathrm{norm}(\cdot, \mathrm{inf})$ *of* MATLAB 7.0 *to get*

$$\frac{\mathrm{norm}(T_f(s) - (T_f(s))_{diag}, \mathrm{inf})}{\mathrm{norm}(T_f(s), \mathrm{inf})} = O(10^{-13}).$$

Therefore, under the measure of the peak gain of the frequency responses, the off-diagonal elements of $T_f(s)$ *are tiny relative to its diagonal elements, and thus* $T_f(s)$ *is diagonal.*

3. Conclusions. We have derived numerically verifiable solvability conditions and developed a numerical algorithm, Algorithm 2, to compute a solution for the state feedback decoupling problem with stability for general proper systems described by (A, B, C, D) quadruples. Algorithm 2 involves only orthogonal transformations and the solutions of several linear systems of equations, and thus it can be implemented in a numerically reliable manner.

Acknowledgments. We would like to thank the anonymous referees and Professor Peter Benner for their helpful suggestions and comments on an early version of this paper.

REFERENCES

[1] P. BENNER AND R. BYERS, *An arithmetic for matrix pencils: Theory and new algorithms*, Numer. Math., 103 (2006), pp. 539–573.
[2] D. CHU AND Y. S. HUNG, *A matrix pencil approach to the row by row decoupling problem for descriptor systems*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 682–702.
[3] D. CHU, M. MALABRE, AND R. C. E. TAN, *State Feedback Decoupling Problem with Stability for* (A, B, C, D) *Quadruples*, Technical Report, Department of Mathematics, NUS, Singapore, 2007.
[4] J. RUIZ-LEON, J. L. OROZCO, AND O. BEGOVICH, *Closed-loop structure of decouplable linear multivariable systems*, Kybernetika (Prague), 41 (2005), pp. 33–45.
[5] D. CHU, L. DE LATHAUWER, AND B. DE MOOR, *A QR-type reduction for computing the SVD of a general matrix product/quotient*, Numer. Math., 95 (2003), pp. 101–121.
[6] Q. G. WANG, *Decoupling Control*, Springer-Verlag, Berlin, 2003.
[7] J. C. ZUNIGA, J. RUIZ-LEON, AND D. HENRION, *Algorithm for decoupling and complete pole assignment of linear multivariable systems*, in Proceedings of the European Control Conference ECC-2003, Cambridge, UK, 2003.
[8] Q. G. WANG AND Y. YANG, *Transfer function matrix approach to decoupling problem with stability*, Systems Control Lett., 47 (2002), pp. 103–110.
[9] D. CHU AND R. C. E. TAN, *Numerically reliable computing for the row by row decoupling problem with stability*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 1143–1170.
[10] P. BENNER AND R. BYERS, *Evaluating products of matrix pencils and collapsing matrix products*, Numer. Linear Algebra Appl., 8 (2001), pp. 357–380.
[11] C. A. SMITH, *Automated continuous process control*, John Wiley, New York, 2001.
[12] R. G. MOREIRA, *Automatic Control for Food Processing Systems*, Aspen Publishers, Gaithersburg, MD, 2001.
[13] D. CHU AND V. MEHRMANN, *Disturbance decoupling for linear time-invariant systems: A matrix pencil approach*, IEEE Trans. Automat. Control, 46 (2001), pp. 802–808.
[14] D. CHU AND V. MEHRMANN, *Disturbance decoupling problem for descriptor systems by state feedback*, SIAM J. Control Optim., 38 (2000), pp. 1830–1858.
[15] G. I. GÓMEZ AND G. C. GOODWIN, *An algebraic approach to decoupling in linear multivariable systems*, Int. J. Control, 73 (2000), pp. 582–599.
[16] M. MOALLEM, R. V. PATEL, AND K. KHORASANI, *Flexible-Link Robot Manipulatiors: Control Techniques and Structural Design*, Springer-Verlag, New York, 2000.

[17] R. Nakashima, M. Ojime, R. Oguro, and T. Tsuji, *A decoupling control method for industrial robots*, in Proceedings of the 6th International Workshop on Advanced Motion Control (March 30–April 1, 2000), Nagoya Institute of Technology, Nagoya, Japan, 2000, pp. 252–257.

[18] F. Ammar-Khodja, A. Bader, and A. Benabdallah, *Dynamic stabilization of systems via decoupling techniques*, ESAIM Control Optim. Calc. Var., 4 (1999), pp. 577–593.

[19] Z. Bai, J. Demmel, and M. Gu, *An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems*, Numer. Math., 76 (1997), pp. 279–308.

[20] C. A. Smith and A. B. Corripio, *Principles and Practice of Automatic Process Control*, 2nd ed., Wiley, New York, 1997.

[21] J. Ruiz-Leon, P. Zagalak, and V. Eldem, *On the Morgan problem with stability*, Kybernetika (Prague), 32 (1996), pp. 425–441.

[22] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[23] T. Williams and P. J. Antsaklis, *Decoupling*, in The Control Handbook, W. S. Levine ed., CRC Press, Boca Raton, FL, 1996, pp. 795–804.

[24] J. C. Martinez Garcia and M. Malabre, *The simultaneous disturbance rejection and regular row by row decoupling with stability: A geometric approach*, IEEE Trans. Automat. Control, 40 (1995), pp. 365–369.

[25] J. C. Martinez Garcia and M. Malabre, *The row by row decoupling problem with stability: A structural approach*, IEEE Trans. Automat. Control, 39 (1994), pp. 2457–2460.

[26] C. Commault, J. M. Dion, and J. Montoya, *Simultaneous decoupling and disturbance rejection: A structural approach*, Internat. J. Control, 59 (1994), pp. 1325–1344.

[27] J. W. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and applications.* I. *Theory and algorithms*, ACM Trans. Math. Software, 19 (1993), pp. 160–174.

[28] J. H. Edward, ed., *Concurrent Engineering: Tools and Technologies for Mechanical System Design*, Springer-Verlag, Berlin, 1993.

[29] P. Zagalak, J. F. Lafay, and A. N. Herrern Hernandez, *The row-by-row decoupling via state feedback: A polynomial approach*, Automatica, 29 (1993), pp. 1491–1499.

[30] G. S. Miminis, *Deflation in eigenvalue assignment of descriptor systems using state feedback*, IEEE Trans. Automat. Control, 38 (1993), pp. 1322–1336.

[31] P. Hr. Petkov, *Perturbation bounds for orthogonal canonical forms and numerical controllability analysis*, IEEE Trans. Automat. Control, 38 (1993), pp. 639–643.

[32] B. Kågström and P. Van Dooren, *A generalized state-space approach for the additive decomposition of a transfer matrix*, Numer. Linear Algebra Appl., 1 (1992), pp. 165–181.

[33] S. K. Madhavan and S. N. Singh, *Inverse trajectory control and zero-dynamics sensitivity of an elastic manipulator*, Int. J. Robotics and Automation, 6 (1991), pp. 179–191.

[34] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, Academic Press, San Diego, CA, 1990.

[35] J. W. Grizzle and A. Isidori, *Block noninteracting control with stability via static state feedback*, Math. Control Signals Systems, 2 (1989), pp. 315–341.

[36] A. De Luca, P. Lucibello, and G. Ulivi, *Inversion techniques for trajectory control of flexible robot arms*, J. Robotic Systems, 6 (1989), pp. 325–344.

[37] T. Williams and P. J. Antsaklis, *A unifying approach to the decoupling of linear multivariable systems*, Int. J. Control, 44 (1986), pp. 181–201.

[38] A. Dickman and R. Sivan, *On the robustness of multivariable linear feedback systems*, IEEE Trans. Automat. Control, 30 (1985), pp. 401–404.

[39] W. M. Wonham, *Linear Multivariable Control: A Geometric Approach*, 2nd ed., Springer-Verlag, New York, 1985.

[40] M. L. J. Hautus and M. Heymann, *Linear feedback decoupling, transfer function analysis*, IEEE Trans. Automat. Control, 28 (1983), pp. 823–832.

[41] D. P. Filev, *Some new results in state space decoupling of multivariable systems.* I. *Link between geometric approach and matrix methods*, Kybernetika (Prague), 18 (1982), pp. 215–233.

[42] D. P. Filev, *Some new results in state space decoupling of multivariable systems.* II. *Extensions to decoupling of systems with $D \neq 0$ and output feedback decoupling*, Kybernetika (Prague), 18 (1982), pp. 330–344.

[43] P. Van Dooren, *The generalized eigenstructure problem in linear system theory*, IEEE Trans. Automat. Control, 26 (1981), pp. 111–129.

[44] J. Descusse, *State feedback decoupling with stability of linear constant (A, B, C, D) quadruples*, IEEE Trans. Automat. Control, 25 (1980), pp. 739–743.

[45] M. Chang and I. B. Rhodes, *Disturbance localization in linear systems with simultaneous decoupling, pole assignment, or stabilization*, IEEE Trans. Automat. Control, 20 (1975), pp. 518–523.
[46] L. M. Silverman and H. J. Payne, *Input–output structure of linear systems with application to the decoupling problem*, SIAM J. Control, 9 (1971), pp. 199–233.
[47] W. M. Wonham and A. S. Morse, *Decoupling and pole assignment in linear multivariable systems: A geometric approach*, SIAM J. Control, 8 (1970), pp. 1–18.
[48] E. G. Gilbert, *The decoupling of multivariable systems by state feedback*, SIAM J. Control, 7 (1969), pp. 50–63.
[49] W. A. Wolovich and P. L. Falb, *On the structure of multivariable systems*, SIAM J. Control, 7 (1969), pp. 437–451.
[50] P. L. Falb and W. A. Wolovich, *Decoupling in the design and synthesis of multivariable control systems*, IEEE Trans. Automat. Control, 12 (1967), pp. 651–659.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1761–1772

INTERVAL GAUSSIAN ELIMINATION WITH PIVOT TIGHTENING*

JÜRGEN GARLOFF†

Abstract. We present a method by which the breakdown of the interval Gaussian elimination caused by division of an interval containing zero can be avoided for some classes of matrices. These include the inverse nonnegative matrices, the totally nonnegative matrices, and the inverse M-matrices—all classes with identically signed inverses. The approach consists of a tightening of the interval pivot by determining the exact range of the pivot over the matrix interval.

Key words. interval Gaussian elimination, inverse nonnegative matrix, totally nonnegative matrix, inverse M-matrix

AMS subject classifications. 65G20, 65G30, 15A48

DOI. 10.1137/080729621

1. Introduction.

Systems of linear interval equations arise when the entries of the coefficient matrix and the right-hand side of a system of linear equations are varying between given bounds; cf. [20, sect. 3.4]. The *solution set* of such a system

$$[A]x = [b],$$

where $[A] = [\underline{A}, \overline{A}]$ is a given n-by-n matrix interval and $[b] = [\underline{b}, \overline{b}]$ is a given vector interval w.r.t. the usual entrywise partial order, is the set

$$\Sigma([A],[b]) := \{x \in \mathbf{R}^n \mid Ax = b,\ A \in [A],\ b \in [b]\}. \tag{1.1}$$

We will assume here throughout that all $A \in [A]$ are nonsingular. We denote the hull of $\Sigma([A],[b])$, i.e., the smallest vector interval containing $\Sigma([A],[b])$, by $[A]^H[b]$. A method to enclose $[A]^H[b]$ is interval Gaussian elimination, which is obtained from the usual (termed *ordinary* henceforth) Gaussian elimination by replacing the real numbers by the related intervals and the real operations by the respective interval operations; see, e.g., [1, Chap. 15], [20, sect. 4.5]. However, interval Gaussian elimination may fail due to division by an interval pivot containing zero, even when ordinary Gaussian elimination works for all matrices $A \in [A]$. There are some classes of interval matrices for which interval Gaussian elimination cannot fail, e.g., the H-matrices; cf. [17].

If ordinary Gaussian elimination is applied without pivoting, the pivots can be represented as the quotient of two successive leading principal minors. This property is used in [18] to modify the interval Gaussian elimination by tightening the interval pivots, and it is shown that the breakdown of interval Gaussian elimination can be avoided in some cases. However, this tightening is obtained by bounding the range of the two leading principal minors independently and then forming the quotient of both enclosures (in addition, the resulting interval is intersected with the ordinary interval

*Received by the editors July 9, 2008; accepted for publication (in revised form) by A. Frommer October 29, 2008; published electronically February 20, 2009. This work was supported by grants from the state of Baden-Württemberg, Germany.
http://www.siam.org/journals/simax/30-4/72962.html
†Faculty of Computer Science, University of Applied Sciences/HTWG Konstanz, Postfach 100543, D-78405 Konstanz, Germany (garloff@htwg-konstanz.de).

pivot). This often causes an overestimation of the range of the pivot. In this paper, we show that for some classes of matrices the exact range of the pivot can be given. As a consequence of the inclusion isotonicity of the interval arithmetic operations, this further tightening of the interval pivot has the additional advantage that the resulting enclosure of $[A]^H[b]$ is not larger than the one obtained by the method of [18] and it is often smaller. Moreover, the range of the pivots can be obtained by running a few ordinary Gaussian elimination procedures in parallel.

The organization of the paper is as follows: In the next section we introduce our notation and recall the interval Gaussian elimination. In section 3 we present pivot tightening. This is applied in section 4 to inverse nonnegative matrices, to totally nonnegative matrices, and to inverse M-matrices. In sections 5 and 6 we show that analogous results can be obtained for a related determinantal function and some algorithms for the solution of structured systems of linear interval equations. We conclude with some remarks in section 7.

2. Notation and interval Gaussian elimination. By $\mathbf{R}^n$, $\mathbf{R}^{n\times n}$, $\mathbf{IR}$, $\mathbf{IR}^n$, and $\mathbf{IR}^{n\times n}$ we denote the set of real vectors with n components, the set of real n-by-n matrices, the set of the compact and nonempty intervals, the set of the intervals of real vectors with n components, and the set of the intervals of real n-by-n matrices, respectively. We also regard vector intervals as interval vectors and matrix intervals as interval matrices, i.e., as vectors and matrices over $\mathbf{IR}$, respectively, and consequently represent them as $[b] = [\underline{b}, \overline{b}] = ([b_i])_{i=1}^n = ([\underline{b}_i, \overline{b}_i])_{i=1}^n$ and $[A] = [\underline{A}, \overline{A}] = ([a_{ij}])_{i,j=1}^n = ([\underline{a}_{ij}, \overline{a}_{ij}])_{i,j=1}^n$. We identify a degenerate interval (vector, matrix) with the (only) real number (vector, matrix) it contains.

We equip $\mathbf{IR}$, $\mathbf{IR}^n$, $\mathbf{IR}^{n\times n}$ with the usual real interval arithmetic; see, e.g., [1, Chap. 10], [20, Chap. 1]. We assume that the reader is familiar with the basic properties of this arithmetic. For a function $f : \mathbf{R}^{n\times n} \to \mathbf{R}$ we denote the range of f over the matrix interval $[A]$ by $f([A])$, i.e.,

$$f([A]) := \{f(A) \mid A \in [A]\}.$$

The interval Gaussian elimination (without pivoting) reads as follows.

Given $[A] \in \mathbf{IR}^{n\times n}$, $[b] \in \mathbf{IR}^n$, define $[A]^{(k)} = ([a_{ij}]^{(k)}) \in \mathbf{IR}^{n\times n}$, $[b]^{(k)} = ([b_i]^{(k)}) \in \mathbf{IR}^n$, $k = 1, \ldots, n$, and $[x]^G = ([x_i]^G) \in \mathbf{IR}^n$ by

$$[A]^{(1)} = [A], \quad [b]^{(1)} = [b],$$

$$[a_{ij}]^{(k+1)} = \begin{cases} [a_{ij}]^{(k)}, & i = 1, \ldots, k,\ j = 1, \ldots, n, \\ [a_{ij}]^{(k)} - \frac{[a_{ik}]^{(k)} \cdot [a_{kj}]^{(k)}}{[a_{kk}]^{(k)}}, & i = k+1, \ldots, n,\ j = k+1, \ldots, n, \\ 0 & \text{otherwise}, \end{cases} \tag{2.1}$$

$$[b_i]^{(k+1)} = \begin{cases} [b_i]^{(k)}, & i = 1, \ldots, k, \\ [b_i]^{(k)} - \frac{[a_{ik}]^{(k)}}{[a_{kk}]^{(k)}} \cdot [b_k]^{(k)}, & i = k+1, \ldots, n, \end{cases}$$

$$k = 1, \ldots, n-1,$$

$$[x_i]^G = \left([b_i]^{(n)} - \sum_{j=i+1}^{n} [a_{ij}]^{(n)} [x_j]^G \right) / [a_{ii}]^{(n)}, \quad i = n, n-1, \ldots, 1,$$

where $\sum_{j=n+1}^{n} \ldots = 0$. Note that no rows or columns are permuted. The algorithm is feasible if and only if $0 \notin [a_{kk}]^{(k)}$, $k = 1, \ldots, n$.

For a survey on results on this algorithm the reader is referred to [17]. Since we are interested only in the feasibility of the algorithm we consider here only the formulae in (2.1) which provide the transformation of $[A]$ into triangular form.

We further adopt standard notation from matrix analysis. For any n-by-n matrix A, we denote the submatrix lying in rows indexed from α and columns indexed from β (both in increasing order) by $A[\alpha \,|\, \beta]$, where $\alpha, \beta \subseteq \{1, 2, \ldots, n\}$. If $\alpha = \beta$, the principal submatrix $A[\alpha \,|\, \alpha]$ is abbreviated as $A[\alpha]$. In particular, when $\alpha = \{1, \ldots, k\}$, $k = 1, \ldots, n$, we call $A[\alpha]$ a *leading* principal submatrix and put $A' := A[\{1, 2, \ldots, n-1\}]$. We denote by A_{ij} the submatrix of A which is obtained by the deletion of row i and column j in A. We call A a *P-matrix* if all its principal minors are positive, *inverse nonnegative*[1] if it is nonsingular and its inverse is entrywise nonnegative, and an M-matrix if it is inverse nonnegative and all its off-diagonal entries are nonpositive. If π is a certain property of a matrix, then we call a matrix interval $[A] = [\underline{A}, \overline{A}]$ a π matrix interval if all $A \in [A]$ possess property π. For example, $[A]$ is an inverse M-matrix interval if all $A \in [A]$ are inverse M-matrices. Often it is of interest to know whether such a property of a matrix interval can be inferred from a certain subset of the *vertex matrices* $A = (a_{ij})$ with $a_{ij} \in \{\underline{a}_{ij}, \overline{a}_{ij}\}$, for all i,j. Examples are given in section 4.

3. Interval pivot tightening. Interval Gaussian elimination breaks down when an interval pivot $[a_{kk}]^{(k)}$ (henceforth termed *ordinary* interval pivot) contains zero. This can occur even though all matrices $A \in [A]$ have nonvanishing leading principal minors [22]. For ordinary Gaussian elimination, the pivot $a_{kk}^{(k)}$ can be represented as the quotient of two succeeding leading principal minors (e.g., [5, p. 26]),

$$a_{kk}^{(k)} = \frac{\det A[\{1, 2, \ldots, k\}]}{\det A[\{1, 2, \ldots, k-1\}]}. \tag{3.1}$$

This property is used in [18] to tighten the ordinary interval pivot $[a_{kk}]^{(k)}$: Let D_k denote an enclosure for $\det A[\{1, 2, \ldots, k\}]$ for all $A \in [A]$, where $D_0 := 1$. If

$$0 \notin D_k, \quad k = 1, 2, \ldots, n, \tag{3.2}$$

then $[a_{kk}]^{(k)}$ in (2.1) is replaced by

$$[a_{kk}]^{(k)} \cap \frac{D_k}{D_{k-1}}. \tag{3.3}$$

In general, this approach does not have any advantage over the ordinary interval pivot because finding an enclosure for the range of a determinant is as difficult as the original problem of solving a system of linear interval equations. However, for some classes of interval matrices presented in section 4 the exact range of principal minors can be given (only by using two specified vertex matrices) and (3.2) is satisfied. Then, when (3.3) is used as the interval pivot the breakdown of interval Gaussian elimination can be avoided. As a welcome side effect, the enclosure of the solution set may be tighter compared to the one which is obtained when the ordinary interval pivot is

[1] This is often also called *inverse positive*; since we use the terminology *totally nonnegative* instead of *totally positive* (cf. subsection 4.2), we consequently prefer *inverse nonnegative*.

used; at least it will not be larger. This is a consequence of the inclusion isotonicity of the interval arithmetic operations.

However, by enclosing the ranges of the two principal minors in (3.1) independently some overestimation is introduced into the computation. In the following, we identify some classes of matrices for which the range of the pivot $a_{kk}^{(k)}$ for all $A \in [A]$ can be given exactly such that forming the intersection in (3.3) is not necessary. Again, as a side effect, this further tightening may result in a further shrinking of the enclosure of the solution set (1.1).

The classes of matrices we will discuss in the next section have the property that any of their leading principal submatrices are in the same class (of lower order). Therefore, the discussion can be restricted to

$$p(A) := \frac{\det A}{\det A'}. \tag{3.4}$$

PROPOSITION 3.1. *The partial derivative of* $p(A) = \det A / \det(A')$ *w.r.t. the entry* a_{ij} *is given by*

$$\frac{\partial p(A)}{\partial a_{ij}} = (-1)^{i+j} \frac{\det A_{nj} \det A_{in}}{(\det A')^2}, \quad i,j = 1,\ldots,n. \tag{3.5}$$

Proof. By Laplacian expansion along row i, we obtain

$$\frac{\partial p(A)}{\partial a_{ij}} = \begin{cases} (-1)^{i+j} \frac{\det A_{ij} \det A' - \det A'_{ij} \det A}{(\det A')^2} & \text{if } i,j < n, \\ (-1)^{i+j} \frac{\det A_{ij}}{\det A'} & \text{if } i = n \text{ or } j = n. \end{cases} \tag{3.6}$$

To show (3.5) for $i,j < n$ we make use of a special case of Sylvester's determinantal identity (e.g., [12, p. 22]): Let $C \in \mathbf{R}^{n\times n}$, $\alpha \subseteq \{1,2,\ldots,n\}$ be a fixed set of cardinality $n-2$, and let $\{1,2,\ldots,n\} \setminus \alpha = \{i,j\}$ with $i < j$. Define the 2-by-2 matrix B by

$$\begin{aligned} b_{11} &:= \det C\,[\alpha \cup \{i\}], \\ b_{12} &:= \det C\,[\alpha \cup \{i\},\, \alpha \cup \{j\}], \\ b_{21} &:= \det C\,[\alpha \cup \{j\},\, \alpha \cup \{i\}], \\ b_{22} &:= \det C\,[\alpha \cup \{j\}]. \end{aligned}$$

Then it holds that

$$\det B = \det C[\alpha] \det C. \tag{3.7}$$

Now, we interchange in A rows 1 and i and columns 1 and j and apply (3.7) to the resulting matrix denoted by C with $\alpha = \{2,3,\ldots,n-1\}$. Since

$$\begin{aligned} \det C[\alpha] &= (-1)^{i+j} \det A'_{ij}, \\ \det C[\{1,\ldots,n-1\}] &= \det A', \\ \det C[\{2,\ldots,n\}] &= (-1)^{i+j} \det A_{ij}, \\ \det C[\{1,\ldots,n-1\} \,|\, \{2,\ldots,n\}] &= (-1)^{j-1} \det A_{nj}, \\ \det C[\{2,\ldots,n\} \,|\, \{1,\ldots,n-1\}] &= (-1)^{i-1} \det A_{in}, \end{aligned}$$

formula (3.7) yields

$$\det A'_{ij} \det A = \det A_{ij} \det A' - \det A_{nj} \det A_{in}.$$

Combining with (3.6), we obtain (3.5) for $i, j < n$. □

Note that $\det A_{nj}$ and $\det A_{in}$ appear in the cofactor form of the inverse of A. Therefore, we consider in the following section sets of matrices with identically signed inverses.

4. Applications.

4.1. Inverse nonnegative matrices. We will make use of the following property of the inverse nonnegative matrices.

PROPOSITION 4.1 (see [16]). *Let $[A] = [\underline{A}, \overline{A}]$ be a matrix interval and $\underline{A}$ and $\overline{A}$ be inverse nonnegative. Then $[A]$ is inverse nonnegative and $\overline{A}^{-1} \leq \underline{A}^{-1}$.*

For results on properties of inverse nonnegative matrix intervals the reader is referred to [23] and for the construction of $[A]^H[b]$ in this case to [21] and the references therein. We will focus here on the applicability of the interval Gaussian elimination. It is known that interval Gaussian elimination may break down for inverse nonnegative interval matrices; see, e.g., [20, p. 160]. We apply the pivot tightening approach from section 3.

We assume now that each $A \in [A]$ has the property that all its leading principal submatrices are inverse nonnegative. By [19, p. 24] this is equivalent to the property that A allows a factorization $A = LDU$, where L and U are inverse nonnegative lower and upper triangular matrices, respectively, whose diagonal entries are all one, and D is a diagonal matrix with positive diagonal entries. According to Proposition 4.1 the condition on $[A] = [\underline{A}, \overline{A}]$ is fulfilled if all leading principal submatrices of $\underline{A}$ and $\overline{A}$ are inverse nonnegative.

THEOREM 4.2. *If all leading principal submatrices of $[A] = [\underline{A}, \overline{A}]$ are inverse nonnegative, then the range of $p(A)$ over $[A]$ is given by*

$$p([A]) = [p(\underline{A}), p(\overline{A})]. \tag{4.1}$$

Proof. For $\underline{A} \leq A \leq \overline{A}$ it follows from Proposition 4.1 that

$$(\overline{A}^{-1})_{nn} \leq (A^{-1})_{nn} \leq (\underline{A}^{-1})_{nn}. \tag{4.2}$$

Formula (4.1) is now a consequence of $(A^{-1})_{nn} = \frac{1}{p(A)}$. □

The practical application of (4.1) requires running in parallel to the interval Gaussian elimination two instances of ordinary Gaussian elimination applied to $\underline{A}$ and $\overline{A}$. In the kth step, both pivots span the interval given on the right-hand side of (4.1).

4.2. Totally nonnegative matrices. A matrix $A \in \mathbf{R}^{n \times n}$ is called *totally nonnegative* or *totally positive* if all its minors are nonnegative or positive, respectively. These matrices appear in mechanics and in many branches of mathematics. If A is nonsingular and totally nonnegative, then so too is each leading principal submatrix and A is a P-matrix. For further properties of these matrices the reader is referred to [3].

A suitable partial order for the totally nonnegative matrices is the *checkerboard order*. For $A, B \in \mathbf{R}^{n \times n}$ define

$$A \leq^* B := (-1)^{i+j} a_{ij} \leq (-1)^{i+j} b_{ij}, \quad i, j = 1, 2, \ldots, n.$$

This partial order is related to the usual entrywise partial order by

$$A \leq^* B \Leftrightarrow A^* \leq B^*, \text{ where } A^* := SAS,\ S := \operatorname{diag}(1, -1, \ldots, (-1)^{n+1}),$$

is the checkerboard transformation.

A matrix interval $[\underline{A}, \overline{A}]$ w.r.t. the usual entrywise partial order can be represented as an interval $[\downarrow A, \uparrow A]^*$ w.r.t. the checkerboard order, where

$$(\downarrow A)_{ij} := \begin{cases} \underline{a}_{ij} & \text{if } i+j \text{ is even,} \\ \overline{a}_{ij} & \text{if } i+j \text{ is odd,} \end{cases}$$

$$(\uparrow A)_{ij} := \begin{cases} \overline{a}_{ij} & \text{if } i+j \text{ is even,} \\ \underline{a}_{ij} & \text{if } i+j \text{ is odd.} \end{cases}$$

It is conjectured in [7][2] that if $\downarrow A$ and $\uparrow A$ are nonsingular and totally nonnegative, then the whole matrix interval $[\downarrow A, \uparrow A]^*$ is nonsingular and totally nonnegative. This conjecture is settled for the totally positive and nonsingular tridiagonal totally nonnegative matrices [7], almost totally positive matrices [10], and interval matrices $[A]$ for which the index sum $i+j$ of all degenerate entries $[a_{ij}]$ has the same parity [7]. In [9] a subset of cardinality of at most 2^{2n-1} vertex matrices is given, from which the total nonnegativity of the entire matrix interval can be inferred.

Note that if A is nonsingular and totally nonnegative, then $0 \leq^* A^{-1}$ and, therefore, $0 \leq (A^{-1})^* = (A^*)^{-1}$. Since A^* is inverse nonnegative, all results for inverse nonnegative matrices carry over to the totally nonnegative matrices by the checkerboard transformation; e.g., if A and B are nonsingular and totally nonnegative, then it follows that $A \leq^* B \Rightarrow B^{-1} \leq^* A^{-1}$. For results on the calculation of $[A]^H[b]$ under special sign conditions posed on $[b]$, cf. [6].

In [6] it was shown that the interval Gaussian elimination may fail if it is applied to a nonsingular totally nonnegative interval matrix. As the most important application of the approach presented in [18], the pivot tightening (3.3) was employed for these interval matrices using the fact that the range of the determinant on such an interval matrix is given by

$$\det A([\downarrow A, \uparrow A]^*) = [\det(\downarrow A), \det(\uparrow A)]. \tag{4.3}$$

However, inspection of (3.5) or the use of the results in subsection 4.1 and application of the checkerboard transformation shows that even the exact range of the pivot of ordinary Gaussian elimination can be given.

COROLLARY 4.3. *If* $[A] = [\downarrow A, \uparrow A]^*$ *is nonsingular and totally nonnegative, then the range of* $p(A)$ *over* $[A]$ *is given by*

$$p([\downarrow A, \uparrow A]^*) = [p(\downarrow A), p(\uparrow A)]. \tag{4.4}$$

Example 4.4. We consider the example in [6], also treated in [18]. Let

$$[A] := \begin{pmatrix} [4,5] & [2,3] & 1 \\ [2,3] & 4 & [2,3] \\ 1 & [2,3] & [4,5] \end{pmatrix}.$$

[2]Note that Theorem 1 in [6] is not correct.

Taking (4.3) into account, it is easily checked that $[A]$ is nonsingular and totally nonnegative. Interval Gaussian elimination results in the interval

$$[a_{33}]^{(3)} = \left[-\frac{79}{700}, \frac{5519}{1280}\right],$$

which contains zero, and breaks down.[3] Using (4.3) the pivot tightening approach from [18] gives for this entry $[a_{33}]^{(3)} \cap \frac{[6,64]}{[7,16]} = [\frac{3}{8}, \frac{5519}{1280}]$, whereas (4.4) results in the smaller interval $[\frac{6}{7}, 4]$.

4.3. Inverse M-matrices. A matrix $A \in \mathbf{R}^{n\times n}$ is called an *inverse M-matrix* if it is the inverse of an M-matrix. Inverse M-matrices are entrywise nonnegative and P-matrices; each leading principal submatrix is likewise an inverse M-matrix. For further properties of these matrices the reader is referred to [13]. Sufficient conditions for a real matrix to be an inverse M-matrix and applications of these matrices can be found in [25].

We will make use of the following proposition.

PROPOSITION 4.5 (see [15]). *A matrix interval is an inverse M-matrix interval if and only if all its vertex matrices are inverse M-matrices.*

Example 4.7 below shows that interval Gaussian elimination may fail if it is applied to such a matrix interval. Pivot tightening here is more involved than in (4.1) and (4.4).

THEOREM 4.6. *If $[A] \in \mathbf{IR}^{n\times n}$ is an inverse M-matrix interval, then the range of $p(A)$ over $[A]$ is given by*

$$p([A]) = \left[p(A^l), p(A^u)\right], \quad \textit{where}$$

$$A^l := \begin{pmatrix} \underline{a}_{11} & \cdots & \underline{a}_{1,n-1} & \overline{a}_{1n} \\ \vdots & & \vdots & \vdots \\ \underline{a}_{n-1,1} & \cdots & \underline{a}_{n-1,n-1} & \overline{a}_{n-1,n} \\ \overline{a}_{n1} & \cdots & \overline{a}_{n,n-1} & \underline{a}_{nn} \end{pmatrix},$$

$$A^u := \begin{pmatrix} \overline{a}_{11} & \cdots & \overline{a}_{1,n-1} & \underline{a}_{1n} \\ \vdots & & \vdots & \vdots \\ \overline{a}_{n-1,1} & \cdots & \overline{a}_{n-1,n-1} & \underline{a}_{n-1,n} \\ \underline{a}_{n1} & \cdots & \underline{a}_{n,n-1} & \overline{a}_{nn} \end{pmatrix}.$$

Proof. According to the sign condition that the inverse of an inverse M-matrix A must have positive diagonal entries and all of its off-diagonal entries must be nonpositive, we obtain

$$\begin{aligned} &0 < \det A_{ii}, \quad i = 1, \ldots, n, \\ &\operatorname{sgn}(\det A_{ij}) \in \{0, (-1)^{i+j+1}\}, \quad i, j = 1, \ldots, n,\ i \neq j. \end{aligned} \tag{4.5}$$

Consider in Proposition 3.1 first the case $i = n$ or $j = n$. It follows that the partial derivative of p w.r.t. the entries in the last row or column is nonpositive (note that

[3] It should be noted that the interval Gaussian elimination would not fail, were $[a_{22}]$ chosen as the first pivot. The same applies to Example 4.7.

$0 < \det A'$) with the exception of a_{nn} for which the partial derivative is positive. In the case $i, j < n$, note that $\operatorname{sgn}(\det A_{nj} \det A_{in}) \in \{0, (-1)^{i+j}\}$. Therefore, the partial derivative of p w.r.t. an entry of A' is always nonnegative. □

Remark 4.6.1. An alternative proof in the case $i, j < n$ is as follows:[4] By permutation similarity of the inverse M-matrices it suffices to consider only entry a_{12}. The inequality

$$\det A_{12} \det A' \leq \det A'_{12} \det A$$

follows from the inequality [14, Theorem 2.1(ii)]

$$(A[\alpha])^{-1} \leq A^{-1}[\alpha]$$

which is valid for inverse M-matrices by choosing $\alpha = \{1, \ldots, n-1\}$.

Example 4.7. Let

$$[A] := \begin{pmatrix} [1,4] & [\frac{1}{2}, \frac{\sqrt{2}}{2}] & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & 1 & [\frac{1}{2}, \frac{\sqrt{2}}{2}] \\ \frac{1}{2} & \frac{\sqrt{2}}{2} & 1 \end{pmatrix}.$$

It is easily checked that all eight vertex matrices are inverse M-matrices, and by Proposition 4.5 it follows that $[A]$ is an inverse M-matrix interval. Interval Gaussian elimination results in the interval

$$[a_{33}^{(3)}] = \left[-\frac{4\sqrt{2}+1}{64}, 1 - \frac{\sqrt{2}}{16} \right],$$

which contains zero, and breaks down. By Theorem 4.6 this pivot can be tightened to $[\frac{1}{2}, 1 - \frac{\sqrt{2}}{4}] = [0.5, 0.6464\ldots]$.

If pivot tightening is applied in all steps, the computation of $p(A^l)$ (and similarly of $p(A^u)$) requires running in parallel to the interval Gaussian elimination an extension of ordinary Gaussian elimination. The elimination algorithm is applied to the submatrices of $[A]$ in the order indicated in Figure 4.1. The part which is recomputed or newly computed is inside the shaded area.

5. A related determinantal function. In a similar way we can determine monotonicity of the related determinantal function

$$d(A) := \det A \det A',$$

which appears, e.g., in the matricial description of the Cholesky decomposition [5, p. 38].

THEOREM 5.1. *Let* $[A] \in \mathbf{IR}^{n \times n}$.

(i) *If* $[A]$ *fulfills the assumption of Theorem* 4.2*, then the range of* $d(A)$ *over* $[A]$ *is given by*

$$d([A]) = [d(\underline{A}), d(\overline{A})].$$

[4] This was pointed out to us by Professor Charles R. Johnson.

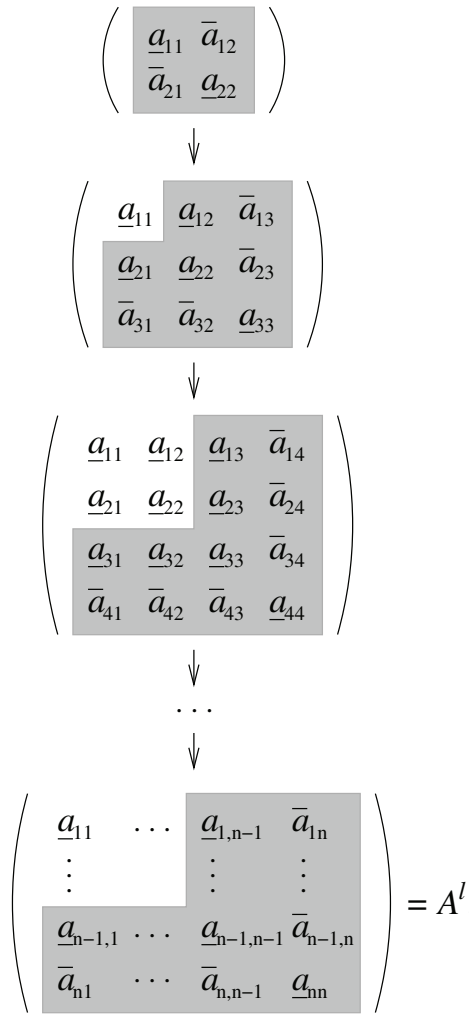


FIG. 4.1. *Elimination algorithm applied to submatrices of* $[A]$.

(ii) *If* $[A]$ *is an inverse* M*-matrix interval, it holds that*

$$d([A]) = [d(A^1), d(A^2)],$$

where

$$A^1 := \begin{pmatrix} \underline{a}_{11} & \overline{a}_{12} & \cdots & \overline{a}_{1n} \\ \overline{a}_{21} & \underline{a}_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \overline{a}_{n-1,n} \\ \overline{a}_{n1} & \cdots & \overline{a}_{n,n-1} & \underline{a}_{nn} \end{pmatrix},$$

$$A^2 := \begin{pmatrix} \overline{a}_{11} & \underline{a}_{12} & \cdots & \underline{a}_{1n} \\ \underline{a}_{21} & \overline{a}_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \underline{a}_{n-1,n} \\ \underline{a}_{n1} & \cdots & \underline{a}_{n,n-1} & \overline{a}_{nn} \end{pmatrix}.$$

Proof. Similarly as in (3.6) we obtain

$$\frac{\partial d(A)}{\partial a_{ij}} = \begin{cases} (-1)^{i+j}(\det A_{ij} \det A' + \det A'_{ij} \det A) & \text{if } i,j < n, \\ (-1)^{i+j} \det A_{ij} \det A' & \text{if } i = n \text{ or } j = n. \end{cases}$$

Let $[A]$ first fulfill the assumption of Theorem 4.2. From the sign condition for the inverse B of an inverse nonnegative matrix A, $B = A^{-1}$, we obtain

$$0 \le b_{ij} = (-1)^{i+j} \frac{\det A_{ji}}{\det A};$$

in particular for $i = j = n$,

$$0 < b_{nn} = \frac{\det A'}{\det A},$$

whence

$$\operatorname{sgn}(\det A) = \operatorname{sgn}(\det A').$$

If $i, j < n$ and $B' := (A')^{-1}$, it follows that

$$\frac{\partial d(A)}{\partial a_{ij}} = (b_{ji} + b'_{ji}) \det A' \det A \geqq 0.$$

If $i = n$ or $j = n$, the proof of the nonnegativity of the partial derivative of d w.r.t. a_{ij} is similar. Therefore, d is monotone increasing. This proves (i).

If $[A]$ is an inverse M-matrix interval, we obtain from the sign condition (4.5) that d is monotone increasing w.r.t. the diagonal entries and monotone decreasing w.r.t. the off-diagonal entries. This proves (ii). ∎

6. Methods for structured systems. The results for interval Gaussian elimination apply to some algorithms for the solution of structured systems of linear interval equations. Interval variants for the respective ordinary methods are obtained by replacing the real numbers by the related intervals and the real operations by the respective interval operations.

If $[A]$ is *symmetric*, i.e., $[A] = [A]^T$, then the *symmetric solution set*

$$\Sigma_{\mathrm{sym}}([A],[b]) := \{x \in \mathbf{R}^n \mid Ax = b,\ A = A^T,\ A \in [A],\ b \in [b]\}$$

is considered. The interval Cholesky method [2] can be used to enclose this set. Each step requires the division by an interval which may contain zero even if all symmetric matrices $A \in [A]$ are positive definite; an example is provided by Example 4.4. For each step of the ordinary Cholesky method, the divisor can be represented as the square root of the quotient of two succeeding leading principal minors as in (3.4) (cf. [5, p. 38]), so that the result for the interval Gaussian elimination applies.

Let $[A]$ be a *Toeplitz* interval matrix (of order $n+1$), i.e., there are intervals $[a_i]$, $i = -n, \ldots, n$, such that $[a_{ij}] = [a_{j-i}]$, $i, j = 1, \ldots, n+1$, and define by $[A]_T$ the set of all real Toeplitz matrices contained in $[A]$. Then one restricts the solution set (1.1) to the *Toeplitz solution set*

$$\Sigma_T([A], [b]) := \{x \in \mathbf{R}^n \mid Ax = b,\ A \in [A]_T, b \in [b]\}.$$

Interval variants of the elimination procedure of Bareiss [4] and of the recurrence relations for the inverse of a Toeplitz matrix [24], [26] are investigated in [8]. Even if all matrices $A \in [A]_T$ have only nonzero leading principal minors, these interval algorithms may break down due to division by an interval containing zero. An example is the interval matrix in [22]. For the ordinary (real) version of both algorithms the divisor can be represented as quotient of two successive leading principal minors (cf. Corollary 1 in [4] and [26, p. 276]), so that for both algorithms the results for the interval Gaussian elimination also apply.

If pivot tightening is used in the symmetric case, then for the three classes of matrix intervals considered in section 4, the respective vertex matrices are symmetric. In the Toeplitz case, the two vertex matrices are Toeplitz matrices in the case of an inverse nonnegative or nonsingular totally nonnegative matrix interval, but in general not in the case of an inverse M-matrix interval.

An elimination process very well suited for totally nonnegative matrices is Neville elimination [11]. Here, zeros in a column below the main diagonal of an n-by-n matrix are produced by adding to each row an appropriate multiple of the previous one (instead of using a fixed row with a fixed pivot as in Gaussian elimination). Lemma 2.6 in [11] shows that all the pivots of Neville elimination are nonzero if and only if the column-initial minors $\det A[\alpha \mid \{1, \ldots, k\}]$, where α is a subset of k successive elements of $\{1, \ldots, n\}$, are nonzero, $k = 1, \ldots, n$. Moreover, in this case, Neville elimination can be carried out without row interchanges. This suggests that the column-initial minors play a role in Neville elimination similar to that of the leading principal minors in Gaussian elimination. Formula (2.8) in [11] shows that the pivots in the kth step of Neville elimination are just the quotients of a column-initial minor of order k and its leading principal minor which is column-initial of order $k-1$. So we can apply Proposition 3.1 and the results from subsection 4.2.

7. Conclusions. We have shown how for some classes of interval matrices the interval pivots in the performance of the interval Gaussian elimination can be tightened such that the shrunken interval does not contain zero, thereby avoiding a breakdown of the algorithm. As a positive side effect, the tightening often leads to a smaller enclosure of the solution set, so that the approach is recommended not only merely for avoiding a breakdown. The extra computational effort consists of two or four instances of ordinary Gaussian elimination, depending on the class of matrices under consideration. However, to obtain verified results, these parallel runs of Gaussian elimination should be performed in interval arithmetic, too. The approach easily extends to some algorithms for solving structured systems of linear interval equations and may be applied to other classes of matrices with identically signed inverses.

Acknowledgments. We thank Professor Charles R. Johnson for his comments on the case of inverse M-matrices and Andrew P. Smith for his careful reading of the manuscript.

REFERENCES

[1] G. Alefeld and J. Herzberger, *Introduction to Interval Computations*, Academic Press, New York, 1983.

[2] G. Alefeld and G. Mayer, *The Cholesky method for interval data*, Linear Algebra Appl., 194 (1993), pp. 161–182.

[3] T. Ando, *Totally positive matrices*, Linear Algebra Appl., 90 (1987), pp. 165–219.

[4] E. H. Bareiss, *Numerical solution of linear equations with Toeplitz and vector Toeplitz matrices*, Numer. Math., 13 (1969), pp. 404–424.

[5] F. R. Gantmacher, *The Theory of Matrices*, Vol. 1, Chelsea Publishing, New York, 1977.

[6] J. Garloff, *Totally nonnegative interval matrices*, in Interval Mathematics 1980, K. Nickel, ed., Academic Press, New York, London, Toronto, 1980, pp. 317–327.

[7] J. Garloff, *Criteria for sign regularity of sets of matrices*, Linear Algebra Appl., 44 (1982), pp. 153–160.

[8] J. Garloff, *Solution of linear equations having a Toeplitz interval matrix as coefficient matrix*, Opuscula Math., 2 (1986), pp. 33–45.

[9] J. Garloff, *Vertex implications for totally nonnegative matrices*, in Total Positivity and Its Applications, M. Gasca and C. A. Micchelli, eds., Kluwer Academic Publishers, Dordrecht, Boston, London, 1996, pp. 103–107.

[10] J. Garloff, *Intervals of almost totally positive matrices*, Linear Algebra Appl., 363 (2003), pp. 103–108.

[11] M. Gasca and J. M. Peña, *Total positivity and Neville elimination*, Linear Algebra Appl., 165 (1992), pp. 25–44.

[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1994.

[13] C. R. Johnson, *Inverse M-matrices*, Linear Algebra Appl., 47 (1982), pp. 195–216.

[14] C. R. Johnson and R. L. Smith, *Almost principal minors of inverse M-matrices*, Linear Algebra Appl., 337 (2001), pp. 253–265.

[15] C. R. Johnson and R. L. Smith, *Intervals of inverse M-matrices*, Reliab. Comput., 8 (2002), pp. 239–243.

[16] J. Kuttler, *A fourth-order finite-difference approximation for the fixed membrane eigenproblem*, Math. Comp., 25 (1971), pp. 237–256.

[17] G. Mayer, *A contribution to the feasibility of the interval Gaussian algorithm*, Reliab. Comput., 12 (2006), pp. 79–98.

[18] J. Mayer, *An approach to overcome division by zero in the interval Gauss algorithm*, Reliab. Comput., 8 (2002), pp. 229–237.

[19] K. Metelmann, *Inverspositive Bandmatrizen und totalnichtnegative Green'sche Matrizen*, Dissertation, University of Cologne, Cologne, Germany, 1972.

[20] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, UK, 1990.

[21] S. Ning and R. B. Kearfott, *A comparison of some methods for solving linear interval equations*, SIAM J. Numer. Anal., 34 (1997), pp. 1289–1305.

[22] K. Reichmann, *Abbruch beim Intervall–Gauss–Algorithmus*, Computing, 22 (1979), pp. 355–361.

[23] J. Rohn, *Inverse-positive interval matrices*, Z. Angew. Math. Mech., 67 (1987), pp. T492–T493.

[24] W. F. Trench, *An algorithm for the inversion of finite Toeplitz matrices*, J. Soc. Indust. Appl. Math., 12 (1964), pp. 515–522.

[25] R. A. Willoughby, *The inverse M-matrix problem*, Linear Algebra Appl., 18 (1977), pp. 75–94.

[26] S. Zohar, *The solution of a Toeplitz set of linear equations*, J. ACM, 21 (1974), pp. 272–276.

SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1773–1788

ON THE CONVERGENCE OF THE SELF-CONSISTENT FIELD ITERATION FOR A CLASS OF NONLINEAR EIGENVALUE PROBLEMS*

CHAO YANG†, WEIGUO GAO‡, AND JUAN C. MEZA†

Abstract. We investigate the convergence of the self-consistent field (SCF) iteration used to solve a class of nonlinear eigenvalue problems. We show that for the class of problems considered, the SCF iteration produces a sequence of approximate solutions that contain two convergent subsequences. These subsequences may converge to two different limit points, neither of which is the solution to the nonlinear eigenvalue problem. We identify the condition under which the SCF iteration becomes a contractive fixed point iteration that guarantees its convergence. This condition is characterized by an upper bound placed on a parameter that weighs the contribution from the nonlinear component of the eigenvalue problem. We derive such a bound for the general case as well as for a special case in which the dimension of the problem is 2.

Key words. nonlinear eigenvalue problem, self-consistent field iteration polynomial

AMS subject classifications. 15A18, 65F15, 47J10

DOI. 10.1137/080716293

1. Introduction.

We are concerned with the convergence of a numerical method for solving the following type of nonlinear eigenvalue problem:

$$H(X)X = X\Lambda_k, \tag{1}$$

where $X \in \mathbb{R}^{n\times k}$, $X^TX = I_k$, $H(X) \in \mathbb{R}^{n\times n}$ is a matrix that has a special structure to be defined below, and $\Lambda_k \in \mathbb{R}^{k\times k}$ is a diagonal matrix consisting of the k smallest eigenvalues of $H(X)$. This type of problem arises in electronic structure calculations [10, 6]. The nonlinearity simply refers to the dependency of the matrix H on the eigenvector X to be computed. This dependency is expressed through a vector $\rho(X)$ that corresponds to the *charge density* of electrons in an electronic structure calculation. This vector is defined as

$$\rho(X) \equiv \mathrm{diag}(XX^T), \tag{2}$$

where $\mathrm{diag}(A)$ denotes the vector containing the diagonal elements of the matrix A.

Given $\rho(X)$, the matrix $H(X)$ that we will consider in this paper is defined as

$$H(X) = L + \alpha \mathrm{Diag}(L^{-1}\rho(X)), \tag{3}$$

where L is a discrete Laplacian, $\mathrm{Diag}(x)$ (with an uppercase D) denotes a diagonal matrix with x on its diagonal, and α is some known constant. In electronic structure calculations, $H(X)$ is often referred to as a single-particle Hamiltonian.

*Received by the editors February 22, 2008; accepted for publication (in revised form) by M. E. Hochstenbach October 20, 2008; published electronically February 25, 2009.
http://www.siam.org/journals/simax/30-4/71629.html

†Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (CYang@lbl.gov, JCMeza@lbl.gov). The work of these authors was supported by the Director, Office of Science, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract DE-AC03-76SF00098.

‡School of Mathematical Sciences, Fudan University, Shanghai, People's Republic of China (wggao@fudan.edu.cn). This author's research was supported by the Special Funds for Major State Basic Research Projects of China (grant 2005CB321700) and the 111 Project.

The solution of (1) is also a global minimizer of the constrained minimization problem

$$\begin{array}{ll} \min & E(X) \\ \text{s.t.} & X^T X = I_k, \end{array} \tag{4}$$

where the objective function $E(X)$ is defined by

$$E(X) = \frac{1}{2}\text{trace}(X^T L X) + \frac{\alpha}{4}\rho(X)^T L^{-1}\rho(X). \tag{5}$$

In fact, it is not difficult to show that (1) and the orthonormality constraint $X^T X = I_k$ form the first order necessary conditions for (4) [7].

The nonlinear eigenvalue problem defined by (1) and (3) is a simplification of the Hartree–Fock (HF) and Kohn–Sham (KS) equations in electronic structure calculations [10, 6]. In particular, it contains a parameterized Hartree term $\rho^T L^{-1}\rho$ that is present in both the HF and KS equations. But it does not contain the exchange term in the HF model [10] or the exchange-correlation term in the KS model [6]. Although our analysis is performed on this simplified model, the main results reveal some of the fundamental properties of this type of problem and how the behavior of the algorithm used to solve this type of problem changes with respect to the amount of nonlinearity measured by the parameter α in (3).

The numerical method we will analyze is called the *self-consistent field* (SCF) iteration. It is currently the most widely used algorithm for solving the HF and KS equations. In each SCF iteration, one computes approximations to a few of the smallest eigenvalues and the corresponding eigenvectors of a fixed Hamiltonian constructed from the previous approximation to X; the computed eigenvector approximations are used to update the Hamiltonian. When the difference between Hamiltonians constructed in two consecutive iterations is negligible, the SCF procedure is terminated, and the eigenvectors of the last Hamiltonian are said to be self-consistent.

It is well known that the simplest version of SCF iteration, which we will carefully describe in the next section, often fails to converge [5]. For certain types of Hamiltonians (e.g., HF and the one defined in (3)), the SCF iteration may eventually oscillate between two limit points, neither of which satisfies (1). The convergence failure of the SCF iteration is partially explained in [11] by viewing the SCF iteration as an indirect minimization procedure that seeks the minimum of (4) by minimizing a sequence of quadratic surrogates. Although the arguments and numerical examples presented in [11] demonstrated that $E(X)$ may not decrease monotonically in an SCF iteration, they do not reveal the asymptotic convergence behavior of the SCF iteration.

In this paper, we will take a closer look at the SCF iteration and analyze its convergence when used to solve (1). A brief overview of the algorithm is given in section 2 along with a simple example that illustrates the convergence failure of the SCF iteration for some choices of α used in (3). In section 3, we show that when the SCF iteration fails to converge, the approximate eigenvectors $X^{(i)}$ produced in the SCF iteration contain two subsequences that converge to two distinct limit points. Neither of these limit points is a solution to (1). Our proof of this result is similar to an earlier proof given by Cancès and Le Bris in [2]. We made a number of simplifications to make it easier to follow. However, the subsequence convergence result does not give the conditions under which the two subsequences are guaranteed to converge to the solution of (1). Such a condition is identified in section 4. We will show that for $n = 2$, the SCF iteration is guaranteed to converge when $\alpha < 3$. For the more general case,

SCF Iteration

Input: A discrete Laplacian $L \in \mathbb{R}^{n\times n}$; an initial guess $X^{(0)}$ for the eigenvector $X \in \mathbb{R}^{n\times k}$;

Output: $X \in \mathbb{R}^{n\times k}$ such that $X^*X = I_k$ and $H(X)X = X\Lambda_k$, where Λ_k contains the k smallest eigenvalues of $H(X)$.

1. For $i = 1, 2, \ldots$ until convergence
2. construct $H^{(i)} = H(X^{(i-1)})$ using (3);
3. compute $X^{(i)}$ such that $H^{(i)}X^{(i)} = X^{(i)}\Lambda^{(i)}$, and $\Lambda^{(i)}$ contains the k smallest eigenvalues of $H^{(i)}$;
4. end for

FIG. 1. *The SCF iteration.*

our main result provides an upper bound for α that depends on the minimum gap between the kth and the $k+1$st eigenvalues of $H(X)$, the dimension of the problem, and the norm of L^{-1}.

Throughout this paper, we will use $\|\cdot\|_p$ to denote the p-norm [3] of either a vector or a matrix. The Frobenius norm of a matrix is denoted by $\|\cdot\|_F$.

2. The SCF iteration. In this section, we describe the SCF iteration and show how it fails when it is applied to a 2×2 Hamiltonian (3) with a particular choice of α.

The basic idea of an SCF iteration is to reduce the nonlinear eigenvalue problem (1) to a sequence of linear eigenvalue problems that can be solved efficiently using existing tools. Figure 1 shows the main steps of this procedure. The convergence of the iteration can be monitored by computing the difference between charge densities $\rho(X)$ obtained in two consecutive iterations. The following example shows that the simplest version of the SCF iteration fails to converge. In this example, we set

$$L = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}, \tag{6}$$

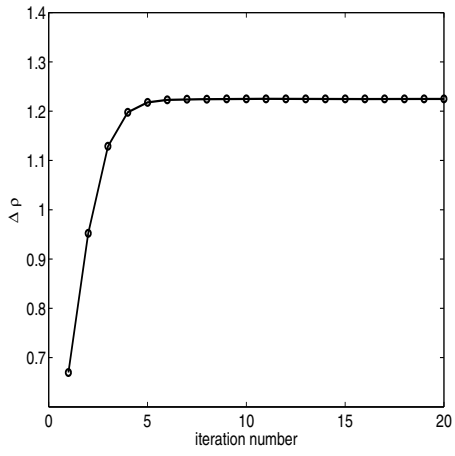
$\alpha = 12$, and $k = 1$. As a result, $X = (x_1 \;\; x_2)^T$ with $x_1, x_2 \in \mathbb{R}$ such that $x_1^2 + x_2^2 = 1$, and $\rho(X) = (x_1^2 \;\; x_2^2)^T$.

Due to the convexity and symmetry of $E(x)$ (i.e., interchanging x_1 and x_2 does not change the problem), the solution to the minimization problem (4), and hence the nonlinear eigenvalue problem (1), must satisfy $x_1 = x_2 = \sqrt{2}/2$ or $x_1 = x_2 = -\sqrt{2}/2$.

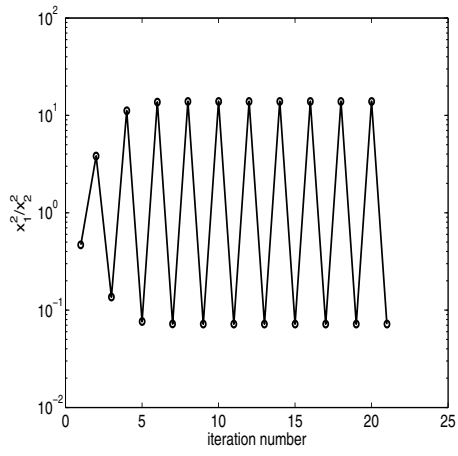
However, when the initial guess of the desired eigenvector is chosen to be, for example,

$$X^{(0)} = \begin{pmatrix} 0.1389 \\ 0.2028 \end{pmatrix}, \tag{7}$$

the difference between the charge densities computed in two consecutive SCF iterations does not converge to zero, as we can clearly see in Figure 2(a). Furthermore, Figure 2(b) shows that the ratio between two components of $\rho(X^{(i)})$ does not converge to 1.



(a) The change in charge density $\Delta\rho^{(i)} \equiv \|\rho(X^{(i+1)}) - \rho(X^{(i)})\|_2$ fails to converge to zero.



(b) The ratio between x_1^2 and x_2^2 oscillates around one, but does not converge to one.

FIG. 2. *The SCF iteration fails to converge when* $\alpha = 12$ *in* (3).

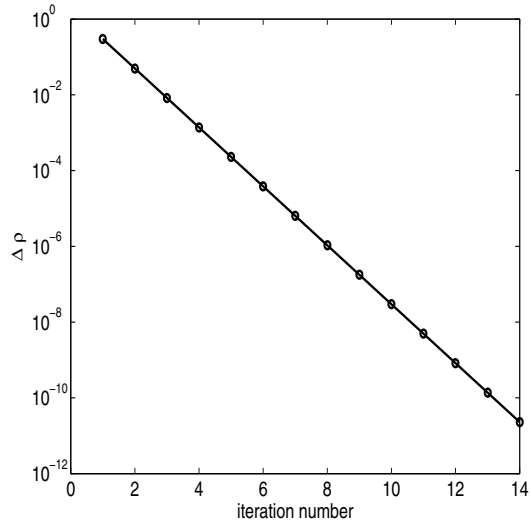
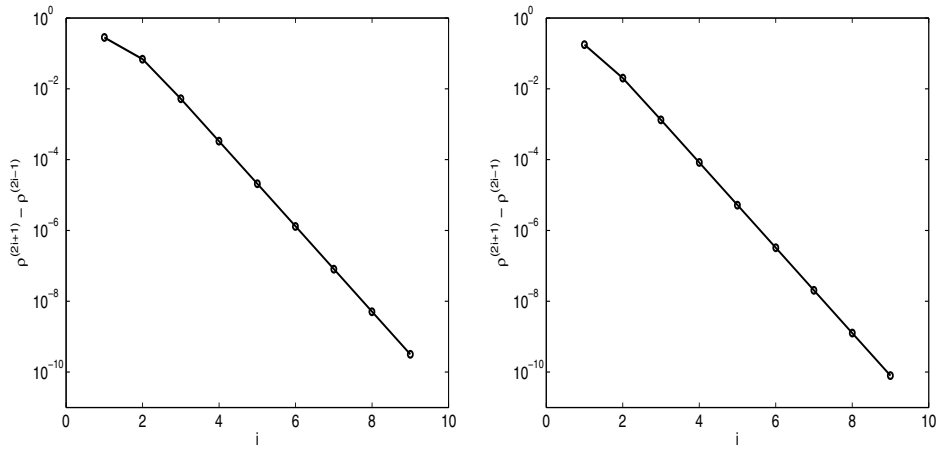


FIG. 3. *When* $\alpha = 1.0$, $\Delta\rho^{(i)}$ *converges rapidly to* 0.

If we reduce α to 1, then SCF converges from any starting guess. Figure 3 shows that the difference between charge densities computed in two consecutive SCF iterations decreases rapidly towards zero in this case when (7) is used as the starting guess. In section 4, we will show that for this 2×2 example, the convergence of SCF can be guaranteed if $\alpha < 3$.

3. Subsequence convergence in the SCF iteration. When the SCF iteration fails to converge to the solution of (1), it produces a sequence of approximations



(a) Odd iterations (b) Even iterations

FIG. 4. *When $\alpha = 12$, the charge density converges to two different limit points in odd and even SCF iterations.*

$\{X^{(i)}\}$ that do not become self-consistent as i increases. We have already seen this phenomenon in Figure 2(a), where we plotted the norm of the change in $\rho(X^{(i)})$ between two consecutive SCF iterations. In this case, it is clear that $\|\Delta\rho(X^{(i)})\|_2$ does not converge to zero as i increases.

However, if we examine the subsequences $\{X^{(2i-1)}\}$ and $\{X^{(2i)}\}$ $(i = 1, 2, \ldots)$ produced in the SCF iteration, we will see that they both converge to subspaces that become self-consistent in every other iteration. Figure 4 shows that both

$$\Delta\rho^{(i)}_{\text{odd}} \equiv \|\rho(X^{(2i+1)}) - \rho(X^{(2i-1)})\|_2 \quad \text{and} \quad \Delta\rho^{(i)}_{\text{even}} \equiv \|\rho(X^{(2i+2)}) - \rho(X^{(2i)})\|_2$$

converge to zero as i increases, although neither $X^{(2i+1)}$ nor $X^{(2i+2)}$ becomes a minimizer of $E(X)$, as we can clearly see in Figure 2(b).

In [1] and [2] it was shown that such a phenomenon occurs in a more general setting; i.e., when SCF fails to converge to the solution of the HF equation, the odd and even subsequences of the approximations converge to two distinct limit points. This analysis, which we will reproduce here with some modification, is based on examining the convergence of the density matrix $D(X) = XX^T$. It relies on the assumption that there exists a gap δ between the kth and $k+1$st eigenvalues of $H(X)$ for all valid X, an assumption that is referred to in [1] as the *uniformly well posed* (UWP) property. The major result of [1] asserts that

$$\sum_{i=\ell}^{\infty} \|D(X^{(i+2)}) - D(X^{(i)})\|_F^2 < \infty$$

for any finite $\ell \geq 0$. Therefore, $\|D(X^{(i+2)}) - D(X^{(i)})\|_F$ must converge to zero as i increases.

In the analysis we present next, the subsequence convergence of the SCF iteration is measured by the distance between two subspaces spanned by columns of $X \in \mathbb{R}^{n\times k}$ and $Y \in \mathbb{R}^{n\times k}$. We will use the standard distance measure defined in

[3, Theorem 2.6.1, p. 76]; i.e., if $X^TX = Y^TY = I_k$,

$$\text{dist}(X, Y) \equiv \|Z^TY\|_2,$$

where $Z \in \mathbb{R}^{n\times(n-k)}$ is the orthogonal complement to X and $Z^TZ = I_{n-k}$.

The following lemma, which is a block version of Lemma 11-9-8 in [8], shows that $\text{dist}(X, Y)$ can, in general, be bounded in terms of $\text{trace}(Y^THY) - \text{trace}(X^THX)$ and the gap between the kth and $k+1$st eigenvalues of H if columns of X consist of eigenvectors associated with the k smallest eigenvalues of H and $Y \in \mathbb{R}^{n\times k}$ satisfies $Y^TY = I_k$.

LEMMA 1. *Let $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ be eigenvalues of a symmetric matrix $H \in \mathbb{R}^{n\times n}$, and let columns of X be eigenvectors associated with $\lambda_1, \lambda_2, \ldots, \lambda_k$. If $\lambda_{k+1} = \lambda_k + \delta$ for some $\delta > 0$, then*

$$\text{dist}^2(X, Y) \le \frac{\text{trace}(Y^THY) - \text{trace}(X^THX)}{\delta} \tag{8}$$

for any $Y \in \mathbb{R}^{n\times k}$ such that $Y^TY = I_k$.

Proof. Let columns $Z \in \mathbb{R}^{n\times(n-k)}$ be eigenvectors associated with $\lambda_{k+1}, \lambda_{k+2}, \ldots, \lambda_n$, and define $\Lambda_k = \text{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_k)$ and $\Lambda_{n-k} = \text{Diag}(\lambda_{k+1}, \lambda_{k+2}, \ldots, \lambda_n)$. It follows from the spectral decomposition of H that

$$\text{trace}(Y^THY) = \text{trace}[(Y^TX)\Lambda_k(X^TY)] + \text{trace}[(Y^TZ)\Lambda_{n-k}(Z^TY)].$$

Since $\lambda_{k+1} = \lambda_k + \delta$, we have $\lambda_i \ge \lambda_k + \delta$ for $i \ge k+1$. Thus,

$$\text{trace}[(Y^TZ)\Lambda_{n-k}(Z^TY)] \ge (\lambda_k + \delta)\|Z^TY\|_F^2.$$

Consequently,

$$\text{trace}(Y^THY) \ge \text{trace}[(Y^TX)\Lambda_k(X^TY)] + \lambda_k\|Z^TY\|_F^2 + \delta\|Z^TY\|_F^2. \tag{9}$$

Because $W = (X, Z)$ defines an orthogonal transformation, we have

$$\|W^TY\|_F^2 = \|Y\|_F^2 = k.$$

Hence

$$\|Z^TY\|_F^2 = \|W^TY\|_F^2 - \|X^TY\|_F^2 = k - \|X^TY\|_F^2. \tag{10}$$

Substituting (10) into (9) and setting $S = X^TY$ yields

$$\begin{aligned}
\text{trace}(Y^THY) &\ge \text{trace}(S\Lambda_kS^T) + \lambda_k(k - \|S\|_F^2) + \delta\|Z^TY\|_F^2 \\
&= \lambda_k k + \text{trace}[S(\Lambda_k - \lambda_kI)S^T] + \delta\|Z^TY\|_F^2 \\
&= \text{trace}(\Lambda_k) + \text{trace}(\lambda_kI - \Lambda_k) + \text{trace}[(\Lambda_k - \lambda_kI)SS^T] + \delta\|Z^TY\|_F^2 \\
&= \text{trace}(X^THX) + \text{trace}[(\lambda_kI - \Lambda_k)(I - SS^T)] + \delta\|Z^TY\|_F^2.
\end{aligned}$$

Because $X^TX = Y^TY = I_k$, the diagonal elements of SS^T are all less than or equal to one. Hence

$$\text{trace}[(\lambda_kI - \Lambda_k)(I - SS^T)] \ge 0.$$

Therefore, we can now conclude that

$$\begin{aligned}\operatorname{trace}(Y^T HY) &\geq \operatorname{trace}(X^T HX) + \delta\|Z^T Y\|_F^2 \\ &\geq \operatorname{trace}(X^T HX) + \delta\|Z^T Y\|_2^2 \\ &= \operatorname{trace}(X^T HX) + \delta \operatorname{dist}^2(X, Y).\end{aligned}$$

Rearranging terms in the above inequality yields (8). ∎

Our analysis of the subsequence convergence will make use of the auxiliary function

$$\hat{E}(X, Y) = \operatorname{trace}(X^T LX) + \operatorname{trace}(Y^T LY) + \alpha\rho(X)^T L^{-1}\rho(Y). \tag{11}$$

This function is similar to the one used in [1], which is defined in terms of density matrices $D(X)$ and $D(Y)$.

It is easy to verify that

$$\rho(X)^T L^{-1}\rho(Y) = \operatorname{trace}(X^T \operatorname{Diag}[L^{-1}\rho(Y)]X) = \operatorname{trace}(Y^T \operatorname{Diag}[L^{-1}\rho(X)]Y).$$

Thus, $\hat{E}(X, Y)$ is clearly symmetric, i.e., $\hat{E}(X, Y) = \hat{E}(Y, X)$, and it can be expressed alternatively as

$$\begin{aligned}\hat{E}(X, Y) &= \operatorname{trace}(X^T H(Y)X) + \operatorname{trace}(Y^T LY) \\ &= \operatorname{trace}(Y^T H(X)Y) + \operatorname{trace}(X^T LX).\end{aligned} \tag{12}$$

We are now ready to show the main result, which we state formally in the following theorem.

THEOREM 1. *Let $X^{(0)} \in \mathbb{R}^{n\times k}$ be the initial guess to the solution of the nonlinear eigenvalue problem* (1) *that satisfies ${X^{(0)}}^T X^{(0)} = I_k$. If columns of $X^{(i)} \in \mathbb{R}^{n\times k}$ contain eigenvectors associated with the smallest k eigenvalues of $H(X^{(i-1)})$, as we would obtain when applying the SCF iteration to* (1)*, and if the gap between the kth and the $k+1$st eigenvalues of $H(X^{(i)})$ is greater than or equal to $\delta > 0$ for all i, then*

$$\sum_{i=0}^{m} \operatorname{dist}^2(X^{(i+2)}, X^{(i)}) \leq \frac{\hat{E}(X^{(0)}, X^{(1)}) - \hat{E}(X^{(m+1)}, X^{(m+2)})}{\delta}, \tag{13}$$

where $\hat{E}(\cdot, \cdot)$ is the auxiliary function defined in (11)*.*

Proof. The proof we give here is similar to that presented in [2]. To simplify notation, we will denote $H(X^{(i+1)})$ by H. Because $X^{(i+2)}$ contains eigenvectors associated with the smallest k eigenvalues of H, it follows from Lemma 1 that

$$\operatorname{trace}({X^{(i+2)}}^T H X^{(i+2)}) + \delta \operatorname{dist}^2(X^{(i+2)}, X^{(i)}) \leq \operatorname{trace}({X^{(i)}}^T H X^{(i)}).$$

Adding $\operatorname{trace}({X^{(i+1)}}^T L X^{(i+1)})$ to both sides of the inequality above and invoking (12) yields

$$\hat{E}(X^{(i+1)}, X^{(i+2)}) + \delta \operatorname{dist}^2(X^{(i+2)}, X^{(i)}) \leq \hat{E}(X^{(i)}, X^{(i+1)}).$$

Rearranging terms in the above inequality yields

$$\operatorname{dist}^2(X^{(i+2)}, X^{(i)}) \leq \frac{\hat{E}(X^{(i)}, X^{(i+1)}) - \hat{E}(X^{(i+1)}, X^{(i+2)})}{\delta}.$$

Summing over i yields the inequality (13). ∎

Because $\hat{E}(X^{(m+1)}, X^{(m+2)})$ can be bounded by a constant for any m, and the left-hand side of (13) is an increasing series, $\operatorname{dist}(X^{(i+2)}, X^{(i)})$ must converge to zero as $i \to \infty$.

4. The convergence of SCF. Although the subsequence convergence analysis characterizes what would happen when the SCF iteration fails to converge, it does not give the conditions under which both the even and odd subsequences are guaranteed to converge to the solution of (1). On the other hand, the numerical examples presented in section 2 appear to indicate that the convergence of SCF for the 2×2 problem depends on the value of α, which weighs the contribution of the nonlinear term $\mathrm{Diag}(L^{-1}\rho(X))$ in the Hamiltonian (3). In this section, we will provide a formal proof that this is indeed true. We will prove that the SCF iteration is guaranteed to converge to the solution of (1) from any starting point when $\alpha < \alpha_{\max}$ for some upper bound $\alpha_{\max}$.

Before we state and derive a general bound for α, we will first examine the convergence of the 2×2 problem shown in section 2 because this problem is relatively easy to analyze and because we can obtain a much tighter upper bound on α in this special case.

In section 4.2, we will use a more sophisticated technique to derive an upper bound for α that is more general but somewhat pessimistic.

4.1. The 2×2 case. Before we get to the main result, we will first show that the ratio between the two components of the charge density oscillates around 1 regardless of the choice of α. We will later show that the magnitude of the oscillation decreases to zero when α is sufficiently small.

LEMMA 2. *Let $y = (y_1 \;\; y_2)^T$ be the eigenvector associated with the smallest eigenvalue of $H(X)$ defined in* (3)*, where $X = (x_1 \;\; x_2)^T$ with $|x_1| > |x_2|$. If $\alpha > 0$ in* (3)*, then $|y_2| > |y_1|$.*

Proof. It is straightforward to write down the inverse of L defined in (6) and show that

$$L^{-1}\rho(X) = \frac{1}{3}\begin{pmatrix} 2x_1^2 + x_2^2 \\ x_1^2 + 2x_2^2 \end{pmatrix}.$$

Consequently, the two diagonal elements in the second term of $H(X)$ in (3) are simply

$$\beta_1 = \frac{\alpha}{3}(2x_1^2 + x_2^2) \quad \text{and} \quad \beta_2 = \frac{\alpha}{3}(x_1^2 + 2x_2^2). \tag{14}$$

Suppose λ is an eigenvalue of $H(X)$; then

$$\det\begin{pmatrix} 2+\beta_1-\lambda & -1 \\ -1 & 2+\beta_2-\lambda \end{pmatrix} = (2+\beta_1-\lambda)(2+\beta_2-\lambda) - 1 = 0. \tag{15}$$

If we let $\phi(\lambda) = (2+\beta_1-\lambda)(2+\beta_2-\lambda)$, then eigenvalues of H are solutions to the equation $\phi(\lambda) = 1$.

It is easy to see from (14) that

$$\beta_1 - \beta_2 = \frac{\alpha}{3}(x_1^2 - x_2^2) > 0, \tag{16}$$

since $|x_1| > |x_2|$. Therefore, the two eigenvalues of $H(X)$, which are distinct roots of the quadratic equation $\phi(\lambda) = 1$, must satisfy

$$\lambda_1 < 2 + \beta_2 < 2 + \beta_1 < \lambda_2. \tag{17}$$

Let $y = (y_1 \;\; y_2)^T$ be the eigenvector associated with λ_1. It follows from $H(X)y = \lambda_1 y$ that

$$(2 + \beta_1 - \lambda_1)y_1 = y_2. \tag{18}$$

Because (17) implies $0 < 2 + \beta_2 - \lambda_1 < 2 + \beta_1 - \lambda_1$, it follows from (15) that

$$2 + \beta_1 - \lambda_1 > 1.$$

Consequently, we can deduce from (18) that $|y_2| > |y_1| > 0$. □

Lemma 3 confirms the observation we made in Figure 2(b), namely, that the ratio between the first and second components of ρ oscillates around 1 in the SCF iteration. The convergence of x_1 and x_2 to the optimal solution can be easily proved if we can show that

$$\frac{|y_2|}{|y_1|} < \frac{|x_1|}{|x_2|} \quad \text{when} \quad |x_1| > |x_2|, \tag{19}$$

or

$$\frac{|y_1|}{|y_2|} < \frac{|x_2|}{|x_1|} \quad \text{when} \quad |x_2| > |x_1|. \tag{20}$$

Without loss of generality, we will establish the condition under which (19) holds. However, before we do that, let us first express y_2/y_1 as a function of $\beta_1 - \beta_2$.

LEMMA 3. *If β_1 and β_2 are defined by* (14)*, then*

$$\frac{y_2}{y_1} = \frac{(\beta_1 - \beta_2) + \sqrt{(\beta_1 - \beta_2)^2 + 4}}{2}, \tag{21}$$

where $y = (y_1, y_2)^T$ is the eigenvector associated with the smallest eigenvalue of $H(X)$.

Proof. Let $\delta = y_2/y_1 = 2 + \beta_1 - \lambda_1$. It is easy to show that

$$2 + \beta_2 - \lambda_1 = \delta - (\beta_1 - \beta_2).$$

Hence, it follows from (15) that

$$\delta^2 - (\beta_1 - \beta_2)\delta - 1 = 0. \tag{22}$$

Solving (22) for δ and taking the positive root yields (21). □

Note that if $x_1 = x_2 = \sqrt{2}/2$, then $\beta_1 - \beta_2 = 0$. In this case, it follows from (21) that $y_2/y_1 = 1$, which matches our intuitive expectation that the SCF iteration should converge right away when the initial guess is the solution to (1).

The following theorem establishes the condition that guarantees the monotonic convergence of the SCF iteration when the initial guess is not the solution to (1).

THEOREM 2. *Let $X = (x_1 \;\; x_2)^T$ be an initial guess of the solution to* (1)*, where $H(X)$ is defined by* (3)*, and let $(y_1 \;\; y_2)^T$ be the eigenvector associated with the smallest eigenvalue of $H(X)$. If $|x_1| > |x_2|$, then*

$$\left|\frac{y_2}{y_1}\right| < \left|\frac{x_1}{x_2}\right| \tag{23}$$

when the parameter α in (3) *satisfies*

$$0 < \alpha \leq 3. \tag{24}$$

Proof. Applying the inequality $\sqrt{(\beta_1 - \beta_2)^2 + 4} \leq (\beta_1 - \beta_2) + 2$ to (21) yields

$$\frac{y_2}{y_1} \leq \beta_1 - \beta_2 + 1.$$

If $|x_1| = 1$ and $x_2 = 0$, then $|y_2/y_1| < \infty = |x_1/x_2|$ for any choice of $\alpha > 0$. Thus (23) certainly holds when α satisfies (24).

If $x_2 \neq 0$, it follows from (16) that

$$\begin{aligned}
\frac{y_2}{y_1} - 1 &\leq \frac{\alpha}{3}(x_1^2 - x_2^2) \\
&= \frac{\alpha}{3}(|x_1| - |x_2|)(|x_1| + |x_2|) \\
&= \frac{\alpha}{3}\left[|x_2|(|x_1| + |x_2|)\right]\left(\left|\frac{x_1}{x_2}\right| - 1\right) \\
&\leq \frac{\alpha}{3}\left(\frac{x_1^2 + x_2^2}{2} + x_2^2\right)\left(\left|\frac{x_1}{x_2}\right| - 1\right) \\
&= \frac{\alpha}{6}(1 + 2x_2^2)\left(\left|\frac{x_1}{x_2}\right| - 1\right).
\end{aligned}$$

Since $x_1^2 + x_2^2 = 1$ and $|x_1| > |x_2|$, x_2^2 must be less than $1/2$. Consequently,

$$\frac{y_2}{y_1} - 1 < \frac{\alpha}{3}\left(\left|\frac{x_1}{x_2}\right| - 1\right).$$

Thus (23) holds if $\alpha \leq 3$. □

The upper bound for α established in Theorem 2 is slightly pessimistic because our experiments show that the SCF iteration converges for α as large as 6.0. However, it is not terribly loose because our experiments also show that convergence failure occurs when $\alpha = 6.5$.

4.2. The more general case. Our analysis of the SCF iteration for the 2×2 problem relies heavily on the symmetry property of the problem and the fact that the solution to the nonlinear eigenvalue problem satisfies $|x_1| = |x_2|$. It is difficult to apply this approach to the more general case in which $n > 2$ and $k > 1$.

Instead of tracking how eigenvectors of $H(X)$ vary from one iteration to another, we will focus in this section on the change in charge density $\rho(X)$. We will use a technique developed in [9] to characterize the mapping between the input charge density used to construct $H(X)$ in (3) and the output charge density obtained directly from the desired eigenvectors of $H(X)$ via (2). We will show that under certain conditions this mapping becomes a contraction when $\alpha < \alpha_{\max}$ for some $\alpha_{\max}$ that depends on the minimum gap between the kth and the $k+1$st eigenvalues of $H(X)$, the norm of L^{-1}, and the problem size n.

We will again assume that there is a gap between the kth and $k+1$st eigenvalues of $H(X)$ for all $X \in \mathbb{R}^{n\times k}$ that satisfies $X^T X = I_k$, and this gap is larger than some lower bound $\delta > 0$. (This is the UWP condition defined in [1].) The significance of this gap will become clear in the following.

Suppose the eigenvalues of $H(X)$ are

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k < \lambda_{k+1} \leq \cdots \leq \lambda_n,$$

for a given X that satisfies $X^T X = I_k$, and the corresponding eigenvectors are $y_1, y_2, \ldots, y_n$. By definition, the density matrix associated with $Y = (y_1, y_2, \ldots, y_k)$ is

$$D(Y) = YY^T.$$

An alternative way to represent this density matrix is

$$D = Z\Omega Z^T,$$

where $Z = (y_1, y_2, \ldots, y_n)$ and $\Omega = \mathrm{Diag}(\underbrace{1, 1, \ldots, 1}_{k}, 0, \ldots, 0)$.

Because $\lambda_k < \lambda_{k+1}$, we can construct a filter function $\phi(\lambda)$ that satisfies

$$\phi(\lambda) = \begin{cases} 1 & \text{for } \lambda = \lambda_1, \lambda_2, \ldots, \lambda_k, \\ 0 & \text{for } \lambda = \lambda_{k+1}, \lambda_{k+2}, \ldots, \lambda_n. \end{cases} \tag{25}$$

If $\phi(\lambda)$ is continuous and differentiable, then we can represent the charge density, which is normally defined as

$$\rho(Y) = \mathrm{diag}(D(Y)),$$

in an alternative form given by

$$\rho = \mathrm{diag}(\phi(H)).$$

If H is constructed from the charge density ρ_{in}, then

$$\rho_{\mathrm{out}} = \mathrm{diag}[\phi(H(\rho_{\mathrm{in}}))]$$

defines a mapping η from ρ_{in} to ρ_{out}, and this is the mapping implicitly constructed at each SCF iteration.

We would like to identify the condition under which η becomes a contraction. Such a condition will ensure that the SCF iteration converges to a fixed point of η that is the solution to our nonlinear eigenvalue problem.

To seek such a condition, we will show that

$$\|\eta(\rho_1) - \eta(\rho_2)\|_1 < \gamma \|\rho_1 - \rho_2\|_1 \tag{26}$$

for any ρ_1 and ρ_2 that satisfy the standard definition (2), and identify the requirement under which $\gamma < 1$.

Constructing a proper filter function is the key to proving (26). We will choose $\phi(t)$ to be a Fermi–Dirac distribution [4] of the form

$$\phi(t) = f_\mu(t) = \frac{1}{1 + e^{\beta(t-\mu)}}, \tag{27}$$

where μ is implicitly determined by the input matrix argument to $\phi(t)$ and $\beta > 0$ is a constant. To be specific, μ is the solution of the equation

$$\mathrm{trace}(\phi(H)) = \mathrm{trace}(f_\mu(H)) = k. \tag{28}$$

Because $\sum_{i=1}^{n} f_\mu(\lambda_i)$ is monotonic with respect to μ for a fixed β, the solution to (28) is unique for any choice of β and H. Figure 5 shows how Fermi–Dirac distributions look with different β values and $\mu = 0$. Notice that a larger β value leads to a sharper drop-off of $\phi(t)$ from 1 to 0.

If the UWP condition holds, then there exists a constant β sufficiently large so that (25) is fulfilled in finite precision arithmetic.

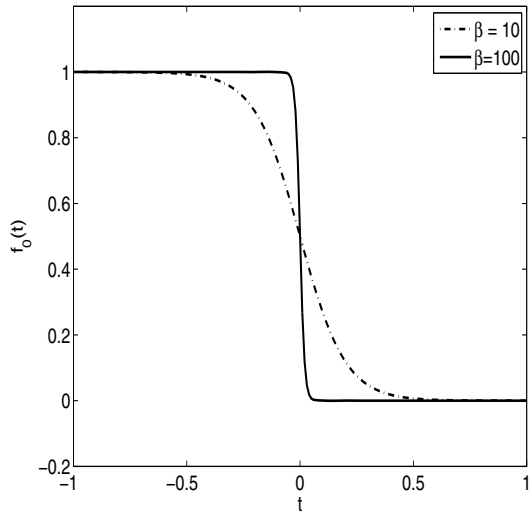


FIG. 5. *Fermi–Dirac distribution* $f_\mu(t) = \frac{1}{1+e^{\beta(t-\mu)}}$ *for* $\mu = 0$.

Let H_1 and H_2 be Hamiltonians constructed from the charge densities ρ_1 and ρ_2, respectively. It is easy to see that

$$\begin{aligned} \|\eta(\rho_1) - \eta(\rho_2)\|_1 &= \|\mathrm{diag}[f_{\mu_1}(H_1)] - \mathrm{diag}[f_{\mu_2}(H_2)]\|_1 \\ &\le \|\mathrm{diag}[f_{\mu_1}(H_1) - f_{\mu_2}(H_1)]\|_1 + \|\mathrm{diag}[f_{\mu_2}(H_1) - f_{\mu_2}(H_2)]\|_1. \end{aligned} \tag{29}$$

Without loss of generality, let us assume $\mu_1 \ge \mu_2$. As a result, $f_{\mu_1}(t) \ge f_{\mu_2}(t)$ for any t. Hence

$$\begin{aligned} \|\mathrm{diag}[f_{\mu_1}(H_1) - f_{\mu_2}(H_1)]\|_1 &= \mathrm{trace}[f_{\mu_1}(H_1) - f_{\mu_2}(H_1)] \\ &= \mathrm{trace}[f_{\mu_1}(H_1)] - \mathrm{trace}[f_{\mu_2}(H_1)]. \end{aligned} \tag{30}$$

Since $\mathrm{trace}[f_{\mu_1}(H_1)] = \mathrm{trace}[f_{\mu_2}(H_2)] = k$, it is easy to see that

$$\begin{aligned} \mathrm{trace}[f_{\mu_1}(H_1)] - \mathrm{trace}[f_{\mu_2}(H_1)] &= \mathrm{trace}[f_{\mu_2}(H_2)] - \mathrm{trace}[f_{\mu_2}(H_1)] \\ &= \mathrm{trace}[f_{\mu_2}(H_2) - f_{\mu_2}(H_1)] \\ &\le \|\mathrm{diag}[f_{\mu_2}(H_2) - f_{\mu_2}(H_1)]\|_1. \end{aligned} \tag{31}$$

Consequently, it follows from (29), (30), and (31) that

$$\begin{aligned} \|\eta(\rho_1) - \eta(\rho_2)\|_1 &\le 2\|\mathrm{diag}[f_{\mu_2}(H_2) - f_{\mu_2}(H_1)]\|_1 \\ &\le 2n\|f_{\mu_2}(H_2) - f_{\mu_2}(H_1)\|_1. \end{aligned} \tag{32}$$

Now to show (26) and to derive an upper bound for α, all we need to do is show that

$$\|f_{\mu_2}(H_2) - f_{\mu_2}(H_1)\|_1 < \frac{\gamma}{2n}\|\rho_1 - \rho_2\|_1$$

for some γ that is proportional to α. Before we do that, we will first prove the following lemma, which allows us to establish a desirable relationship between $f_{\mu_2}(H_2) - f_{\mu_2}(H_1)$ and $H_2 - H_1$.

LEMMA 4. *Let $A, B \in \mathbb{R}^{n\times n}$ be two symmetric matrices, and let $f(t)$ be the Fermi–Dirac distribution defined in* (27). *Suppose $A = V_A D_A V_A^T$ and $B = V_B D_B V_B^T$ are the spectral decompositions of A and B, respectively, i.e., $V_A^T V_A = V_B^T V_B = I$ and*

$$D_A = \begin{pmatrix} \lambda_1^A & & & \\ & \lambda_2^A & & \\ & & \ddots & \\ & & & \lambda_n^A \end{pmatrix}, \quad D_B = \begin{pmatrix} \lambda_1^B & & & \\ & \lambda_2^B & & \\ & & \ddots & \\ & & & \lambda_n^B \end{pmatrix}.$$

Then the identity

$$f(A) - f(B) = V_A(C \odot \Delta)V_B^T$$

holds, where $\Delta = V_A^T(A - B)V_B$, the (j,k)th entry of the matrix C is defined by

$$C_{j,k} = \begin{cases} \frac{f(\lambda_j^A) - f(\lambda_k^B)}{\lambda_j^A - \lambda_k^B} & \text{if } \lambda_j^A \neq \lambda_k^B, \\ f'(\lambda) & \text{if } \lambda_j^A = \lambda_k^B = \lambda, \end{cases}$$

and $C \odot \Delta$ denotes the Hadamard product of C and Δ.

Proof. It follows from the matrix version of the Cauchy integral formula [3] that

$$f(A) - f(B) = \frac{1}{2\pi i} \oint_\Gamma f(z) \Big[(zI - A)^{-1} - (zI - B)^{-1} \Big] dz, \tag{33}$$

where Γ is a closed contour that contains the spectra of both A and B.

Using the identity

$$(zI - A)^{-1} - (zI - B)^{-1} = (zI - A)^{-1}(A - B)(zI - B)^{-1},$$

we can express the right-hand side of (33) as

$$\begin{aligned} &\frac{1}{2\pi i} \oint_\Gamma f(z) V_A (zI - D_A)^{-1} V_A^T (A - B) V_B (zI - D_B)^{-1} V_B^T dz \\ &\quad = \frac{1}{2\pi i} \oint_\Gamma f(z) V_A [(w_A(z) w_B(z)^T) \odot \Delta] V_B^T dz, \end{aligned} \tag{34}$$

where $w_A = \mathrm{diag}[(zI - D_A)^{-1}]$, $w_B = \mathrm{diag}[(zI - D_B)^{-1}]$.

Since the only term in (34) that contains z is $w_A(z)w_B(z)^T$, it follows that

$$f(A) - f(B) = V_A \left[\left(\frac{1}{2\pi i} \oint_\Gamma f(z) w_A(z) w_B(z)^T dz \right) \odot \Delta \right] V_B^T.$$

Let

$$C = \frac{1}{2\pi i} \oint_\Gamma f(z) w_A(z) w_B(z)^T dz.$$

It is easy to verify that the (j,k)th entry of C can be expressed as

$$C_{j,k} = \frac{1}{2\pi i} \oint_\Gamma \frac{f(z)}{(z - \lambda_j^A)(z - \lambda_k^B)} dz. \tag{35}$$

If $\lambda_j^A \neq \lambda_k^B$, the expression above can be evaluated as

$$C_{j,k} = \frac{1}{2\pi i}\frac{1}{\lambda_j^A - \lambda_k^B}\oint_\Gamma \left(\frac{f(z)}{z-\lambda_j^A} - \frac{f(z)}{z-\lambda_k^B}\right)dz. \tag{36}$$

If $\lambda_j^A = \lambda_k^B = \lambda$, (35) becomes

$$C_{j,k} = \frac{1}{2\pi i}\oint_\Gamma \frac{f(z)}{(z-\lambda)^2}dz. \tag{37}$$

Invoking the scalar version of the Cauchy integral formula in both (36) and (37), we then obtain

$$C_{j,k} = \begin{cases} \frac{f(\lambda_j^A)-f(\lambda_k^B)}{\lambda_j^A-\lambda_k^B} & \text{if } \lambda_j^A \neq \lambda_k^B, \\ f'(\lambda) & \text{if } \lambda_j^A = \lambda_k^B = \lambda. \end{cases} \qquad \square$$

Suppose $H_1 = X_1\Lambda_1 X_1^T$ and $H_2 = X_2\Lambda_2 X_2^T$ are the spectral decompositions of H_1 and H_2, respectively. A direct application of Lemma 4 to H_1 and H_2 yields

$$\begin{aligned} \|f_{\mu_2}(H_2) - f_{\mu_2}(H_1)\|_1 &= \|X_2[C \odot (X_2^T(H_2-H_1)X_1)]X_1^T\|_1 \\ &\leq n\|C \odot (X_2^T(H_2-H_1)X_1)\|_1 \\ &\leq n^2\|C\|_1\|H_2-H_1\|_1 \\ &\leq \alpha n^2\|C\|_1\|L^{-1}\|_1\|\rho_2-\rho_1\|_1. \end{aligned} \tag{38}$$

To establish an upper bound for $\|C\|_1$, we can use the mean value theorem and the fact that

$$|f'_\mu(t)| = \left|\frac{-\beta e^{\beta(t-\mu)}}{(1+e^{\beta(t-\mu)})^2}\right| \leq \frac{\beta}{4}$$

to first show that

$$\max_{j,k}|C_{j,k}| \leq \beta/4.$$

It follows immediately that

$$\|C\|_1 \leq n\beta/4. \tag{39}$$

Combining (32), (38), and (39), we obtain

$$\|\eta(\rho_2) - \eta(\rho_1)\|_1 \leq \frac{\alpha n^4\beta\|L^{-1}\|_1}{2}\|\rho_2-\rho_1\|_1.$$

We can easily see that η is a contraction if α satisfies

$$\alpha < \frac{2}{n^4\beta\|L^{-1}\|_1}. \tag{40}$$

It may seem surprising that the upper bound that ensures $\eta(\rho)$ becomes a contraction depends on a parameter β that is present in neither the original eigenvalue problem (1) nor the description of the SCF iteration. However, if we go back to Figure 5 and recall that the choice of β is dictated by the smallest gap between $\lambda_k(H)$ and

$\lambda_{k+1}(H)$ for all valid H matrices, then it becomes clear that the dependency of (40) on β simply says that for problems in which the gap between $\lambda_k(H)$ and $\lambda_{k+1}(H)$ is small, a smaller upper bound of α is required to ensure that the SCF iteration converges from any starting point.

We should point out that the bound established in (40) is pessimistic. In particular, the n^4 factor on the denominator, which is introduced by the use of a loose inequality in (32) and the use of 1-norms to bound the norms of the orthogonal matrices X_1 and X_2 in (38), is rather conservative. In our numerical experiments, we observed that the SCF iteration may converge for α values that are much larger than the right-hand side of (40). However, the qualitative behavior of the SCF iteration seems to be correctly characterized by (40). Table 1 shows both the experimentally observed largest α values (α_1) for which the SCF iteration converges and the experimentally observed smallest α values (α_2) for which the SCF iteration fails to converge for problems with different choices of n and k. The optimal bound lies within the interval (α_1, α_2). We can clearly see that the optimal bound for α decreases as n increases. For the same value of n, changing the value of k in Table 1 results in a change of the gap $\lambda_{k+1} - \lambda_k$. For each combination of n and k, the smallest gap among the various choices of α's that we experimented with is shown in Table 1. The last two rows of Table 1 clearly indicate that for the same n, a smaller $\lambda_{k+1} - \lambda_k$, which corresponds to a larger β value in (40), leads to a more restrictive choice of α for which the SCF iteration is guaranteed to converge.

TABLE 1

Observation from numerical experiments performed to determine the optimal bound for α. In these experiments, the L matrix in (3) *is constructed as the one-dimensional discrete Laplacian with* 2 *on the diagonal and* -1 *on the sub- and sup-diagonals. The dimension of the matrix is n. We look for k smallest eigenvalues and the corresponding eigenvectors. The SCF iteration converges for $\alpha \leq \alpha_1$ and fails to converge for $\alpha \geq \alpha_2$. This implies that the optimal bound for α lies in (α_1, α_2). The spectral gap $\lambda_{k+1} - \lambda_k$ listed here is smallest among all gaps associated with the different choices of α values that we experimented with. These gaps were computed using a trust-region enabled SCF iteration discussed in* [11].

n	k	$\lambda_{k+1} - \lambda_k$	$\|L^{-1}\|_1$	α_1	α_2
2	1	2.0	1.0	6.0	6.5
10	2	0.37	15.0	0.8	0.9
100	10	0.02	1275.0	0.05	0.06
100	4	0.0087	1275.0	0.002	0.0025

In general, the minimum gap between $\lambda_k(H)$ and $\lambda_{k+1}(H)$ is not known a priori. However, when α is sufficiently small, we can estimate such a gap by calculating the difference between the kth and $k+1$st eigenvalues of L. Such an estimate can in turn be used to derive a suitable β value that would allow (27) to achieve the filtering effect (25) in finite precision arithmetic.

5. Concluding remarks. We examined the convergence of the self-consistent field (SCF) iteration used to solve a class of nonlinear eigenvalue problems defined in (1). Our analysis shows that for this type of problem the SCF iteration produces a sequence of approximate solutions $X^{(i)}$ that contain two convergent subsequences. However, the limit points associated with these convergent subsequences may be different, as we demonstrated in a numerical example. We identified the condition under which the SCF iteration becomes a contractive fixed point iteration that will converge to the solution of the nonlinear eigenvalue problem. Our main result suggests that this condition can be characterized by an upper bound placed on the parameter α in (1).

In the most general case, the upper bound we derived characterizes the qualitative behavior of the SCF iteration, although the bound itself is somewhat pessimistic. When the dimension of the problem is 2×2, we can give a much tighter bound using a completely different technique. To generalize such a bound for the Hartree–Fock (HF) or the Kohn–Sham (KS) problem, we need to analyze the relative contribution of the exchange and exchange-correlation terms to the HF and KS Hamiltonians, respectively. We will pursue such analysis in future research.

Acknowledgments. We would like to thank the anonymous referees and the associate editor for careful reading and helpful comments.

REFERENCES

[1] C. Le Bris, *Computational chemistry from the perspective of numerical analysis*, Acta Numer., 14 (2005), pp. 363–444.
[2] E. Cancès and C. Le Bris, *On the convergence of SCF algorithms for the Hartree-Fock equations*, Math. Model. Numer. Anal., 34 (2000), pp. 749–774.
[3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
[4] C. Kittel and H. Kroemer, *Thermal Physics*, W. H. Freeman, San Francisco, 1980.
[5] J. Koutecký and V. Bonacic, *On convergence difficulties in the iterative Hartree-Fock procedure*, J. Chem. Phys., 55 (1971), pp. 2408–2413.
[6] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press, Cambridge, UK, 2004.
[7] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.
[8] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[9] E. Prodan and P. Nordlander, *On the Kohn–Sham equations with periodic background potentials*, J. Statist. Phys., 111 (2003), pp. 967–992.
[10] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: An Introduction to Advanced Electronic Structure Theory*, Dover, New York, 1996.
[11] C. Yang, J. C. Meza, and L.-W. Wang, *A trust region direct constrained minimization algorithm for the Kohn–Sham equation*, SIAM J. Sci. Comput., 29 (2007), pp. 1854–1875.